DUDLEY KNOX LIBRARY
NAVAL POSTGRADUATE SCHOOL
MONTEREY CA 93943-5101

NAVAL POSTGRADUATE SCHOOL
Monterey, California

THESIS

STABILITY AND CONTROL MODULE FOR JOINT ARMY/NAVY ROTORCRAFT ANALYSIS AND DESIGN (JANRAD) SOFTWARE AND GRAPHICAL USER INTERFACE (GUI)

by

David A. Heathorn

March 1999

Thesis Advisor: E. Roberts Wood
Co-Advisor: Robert L. King

Approved for public release; distribution is unlimited.

REPORT DOCUMENTATION PAGE

Form Approved OMB No. 0704 0188

1. AGENCY USE ONLY *(Leave blank)*	2. REPORT DATE March 1998	3. REPORT TYPE AND DATES COVERED Master's Thesis

4. TITLE AND SUBTITLE : STABILITY AND CONTROL MODULE FOR JOINT ARMY/NAVY ROTORCRAFT ANALYSIS AND DESIGN (JANRAD) SOFTWARE AND GRAPHICAL USER INTERFACE (GUI)	5. FUNDING NUMBERS
6. AUTHOR(S) Heathorn, David A.	
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) Naval Postgraduate School Monterey, CA 93943 5000	8. PERFORMING ORGANIZATION REPORT NUMBER
9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES)	10. SPONSORING / MONITORING AGENCY REPORT NUMBER

11. SUPPLEMENTARY NOTES

The views expressed in this thesis are those of the author and do not reflect the official policy or position of the Department of Defense or the U.S. Government.

12a. DISTRIBUTION / AVAILABILITY STATEMENT	12b. DISTRIBUTION CODE
Approved for public release; distribution is unlimited.	

13. ABSTRACT The Joint Army/Navy Rotorcraft Analysis and Design (JANRAD) computer program has been developed at the Naval Postgraduate School to aid in the preliminary design of rotorcraft and has been updated to include a Graphical User Interface (GUI). This thesis is a continuation of the program focusing on stability and control analysis. The trim solution for a specified flight condition is computed from the Performance module of the program. This trim solution is then used to compute stability derivatives for the specified flight condition and a linear state space model is created. This solution can then be used to perform various time and frequency domain analyses or can be saved to a file for future use.

14. SUBJECT TERMS Stability, Control, Design, Rotorcraft, Linear, Modeling, Helicopter, JANRAD	15. NUMBER OF PAGES 268
	16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
Unclassified	Unclassified	Unclassified	UL

NSN 7540 01 280 5500
Standard Form 298 (Rev. 2 89)

Approved for public release; distribution is unlimited

STABILITY AND CONTROL MODULE FOR JOINT ARMY/NAVY ROTORCRAFT ANALYSIS AND DESIGN (JANRAD) SOFTWARE AND GRAPHICAL USER INTERFACE (GUI)

David A. Heathorn
Lieutenant, United States Navy
B.S., University of Colorado at Boulder, 1990

Submitted in partial fulfillment of the
requirements for the degree of

MASTER OF SCIENCE IN AERONAUTICAL ENGINEERING

from the

NAVAL POSTGRADUATE SCHOOL
March 1999

ABSTRACT

The Joint Army/Navy Rotorcraft Analysis and Design (JANRAD) computer program has been developed at the Naval Postgraduate School to aid in the preliminary design of rotorcraft and has been updated to include a Graphical User Interface (GUI). This thesis is a continuation of the program, focusing on stability and control analysis. The trim solution for a specified flight condition is computed from the Performance module of the program. This trim solution is then used to compute stability derivatives for the specified flight condition and a linear state space model is created. This solution can then be used to perform various time and frequency domain analyses or can be saved to a file for future use.

DISCLAIMER

Readers are cautioned that the computer code in this thesis may not have been exercised for all cases of interest. While effort has been made, within the time available, to ensure that the program is free of computational and logical error, additional verification should be applied. This version of JANRAD was written and tested in MATLAB® version 5.0 Student Edition. The use of this application is at the risk of the user.

TABLE OF CONTENTS

LIST OF FIGURES

ACKNOWLEDGMENTS

The author would like to acknowledge Prof. E. Roberts Wood for his insight, and enthusiasm for this program and for his love of all things related to the helicopter. To LCDR Robert King for his in depth knowledge of MATLAB and the original JANRAD program. To LT Pete Tyson for his program on the handling qualities of helicopters. Finally, and most importantly, to my wife who was busy at home with our children while I was laboring over this project.

I. INTRODUCTION

A. BACKGROUND

The Joint Army/Navy Rotorcraft Analysis and Design (JANRAD) computer program was originally developed by students at the Naval Postgraduate School (NPS) as an aid to design for the 1993 American Helicopter Association (AHS) Design Competition. It has since undergone six upgrades.

The original JANRAD developed by Robert Kirk Nicholson, Jr. [Ref. 1] was a command line type program written in MATLAB®, that was used for performance analysis. Inputs for the configuration of the proposed helicopter were entered via the command line and a trim solution for a specified flight condition was obtained. Central to the performance analysis is the harmonic balance method of Gerstenberger and Wood [Ref. 2]. Immediately following Nicholson's work was the update created by Walter M. Wirth [Ref. 3], which provided a linear state space representation for the helicopter that could be used for stability and control analysis. The third version by Juan D. Cuesta [Ref. 4] and revised by Daniel S. Hiatt [Ref. 5] added a Blade Dynamics module to JANRAD utilizing a modified Myklestad Prohl Transfer Matrix Method. These four versions of JANRAD were validated by David M. Eccles [Ref. 6] through comparison with NASA-Army H-34 flight test data (1964) and more recent NASA-Army UH-60A flight tests at NASA Ames (1992).

The next generation JANRAD Version 5.0 was introduced by Chris F. Lapacik [Ref. 7] and revised by William L. Hucke [Ref. 8]. This version brought the JANRAD

program up to date with the Microsoft® Windows® operating environment through the use of MATLAB® version 5.0 Student Edition. The Performance Module of JANRAD was updated with a Graphical User Interface (GUI) and several output plots were generated and verified.

The focus of this thesis was intended to be to add the GUI to the Stability and Control and Rotor Dynamics modules of JANRAD. The input screens for the Stability and Control Module were developed with the help of William L. Hucke. The intention was simply to attach this as the "front end" to the Stability and Control code written by Walter M. Wirth, Jr. [Ref. 3], and then do the same for the Rotor Dynamics Module. Due to various problems encountered with the existing Stability and Control code, the Rotor Dynamics Module did not get updated to the GUI. Instead, several output screens were developed for the Stability and Control Module. The aim was to create output that could be easily applied to the Aeronautical Design Standard Handling Qualities Requirements for Military Rotorcraft (ADS-33), the military's accepted standard for helicopter design. The intent of the release of ADS-33 was to replace MIL-H-8501, which had been the standard since the late 1950's.

B. ORIGINAL STABILITY AND CONTROL CODE

The original code for the Stability and Control module written by Walter M. Wirth, Jr. [Ref. 3] could only be verified for the hover flight regime (0 - 40 knots). The output matrices for the forward flight case were inconsistent with the known values given for Prouty's example helicopter [Ref. 9].

Wirth's code [Ref. 3] utilized the hover trim condition generated by the Performance Module as the starting point for his computations. This process will briefly be discussed in a later section. Stability derivatives were determined by using closed form solutions when possible and by solving multiple trim solutions about a nominal point to solve for unknowns. This method proved effective for the case of the hovering aircraft, where perturbations were performed by varying gross weight. However, when the forward flight condition was evaluated, new stability derivatives were needed that could not be correctly evaluated by simply solving multiple trim solutions about a nominal point. These particular derivatives where not used in the hover case and therefore did not effect the solution for this regime. The bulk of this thesis was spent determining an acceptable method to determine these stability derivatives.

C. USER'S GUIDE

A User's Guide is attached as Appendix A and gives an overview of the major features and procedures for using JANRAD version 6.0. It is a compilation of several thesis students work and therefore is presented in its entirety. It has been updated to reflect the changes and improvements to the Stability and Control Module.

D. JANRAD VERSION 6.0 FILE STRUCTURE AND FLOW CHART

An updated flow chart which tracks files and Callback operation of JANRAD version 6.0 Stability and Control Module is included as Figure 1. The files referenced in the flow chart are included as appendices. Asterisked files are files that were written during prior versions and were not updated by the author.

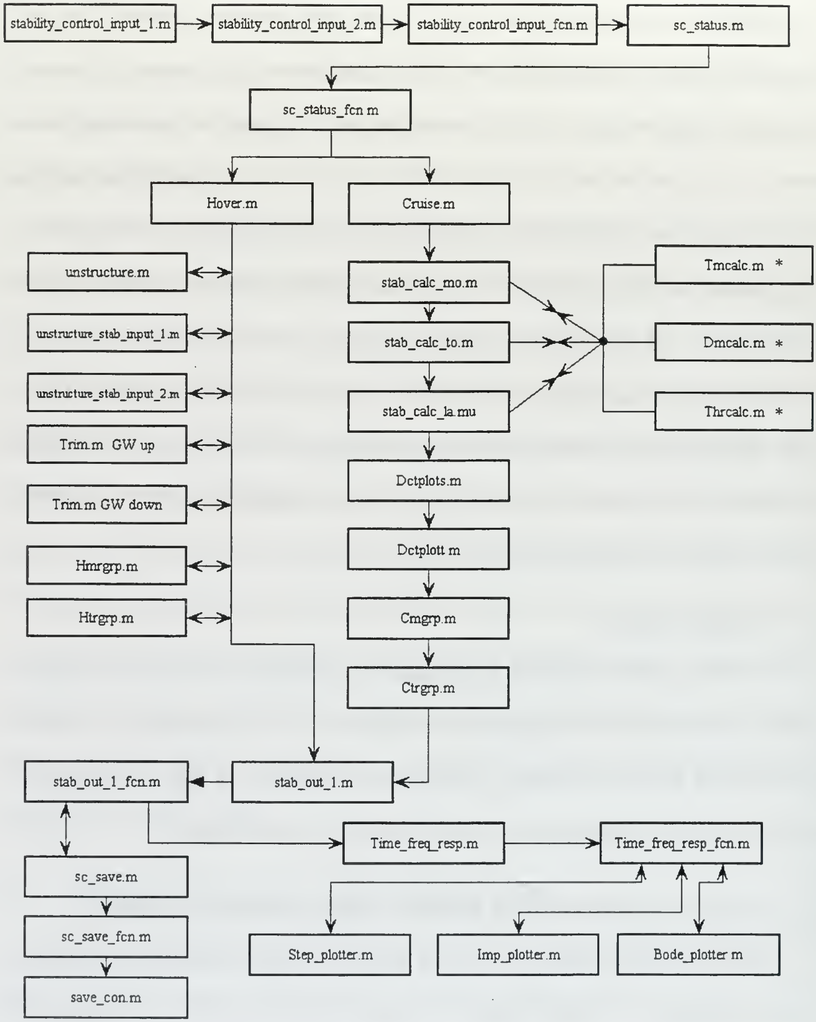

Figure 1. JANRAD Stability and Control Program Architecture

II. INPUT SCREENS

A. GRAPHICAL USER INTERFACE

The original versions of JANRAD where written in MATLAB® PC version 3.5. Entries were made to the program on the command line in the workspace. The process was slow and cumbersome, and did not allow the user to easily change parameters or view entries and outputs. With the development of the GUI, data entry has become easy and intuitive. Input parameters can be entered, changed, and saved with the touch of a button.

The Performance module was updated with the GUI by Chris F. Lapacik [Ref. 7] and modified by William H. Hucke [Ref. 8]. A detailed explanation of the Guide® function in MATLAB® can be found in Lapacik's thesis [Ref. 7]. This function allows for GUI screen to be created using a drag and drop technique. The function writes all of the computer code necessary to create the desired screen. The developer needs only to attach the necessary functions to the screen code. Also in Lapacik's thesis [Ref. 7] is a detailed discussion of the variable and file naming structures. Every attempt has been made to maintain these conventions.

B. STABILITY AND CONTROL INPUT SCREENS

The Stability and Control input screens were developed in conjunction with William L. Hucke, and served as an invaluable learning tool and "pass down". During this process, the general structure of the JANRAD program was learned and instruction

was given as to the use of the GUIDE® function in MATLAB®. An explanation of the development of these screens can be found in Hucke's thesis [Ref. 8].

Once the input screens were created, the variables entered were structured so they could be passed to the appropriate M-files for computation. The program was then tied to the portion of JANRAD written by Walter M. Wirth [Ref. 3] for the stability derivative computation.

Output screens were created to display the linear matrix state equation A and B matrices and a time - frequency analysis screen was created.

III. HOVER FLIGHT CONDITION

A. THE EXISTING CODE

The existing JANRAD Stability and Control Module produced a linear state space model of the helicopter. It utilized the trim solution from the Performance module as the starting point for its calculations and then varied gross weight to obtain the necessary stability derivatives. A detailed explanation of the method used can be found in Wirth's thesis [Ref. 3].

B. MODIFICATIONS

The only modifications to this portion of the program were to add the graphical user interface to the input screens and create a screen to monitor the progress of the program. The input screens were created in conjunction with William Hucke and served as an effective means of learning the intricacies of the MATLAB program. Once the screens were created the variables were structured and passed to the subroutines created by Walter M. Wirth [Ref. 2].

A progress screen was created to aid the user in determining what the program was doing and also serves to ensure that the program has not "locked up" on a particular calculation. This screen is similar to the screen created in the Performance Module.

Once the stability derivatives have been obtained, the A and B matrices are constructed. The matrices obtained in this manner seemed to agree well with that of the example helicopter given in Prouty's book. A detailed comparison of the output

generated by JANRAD and that given by Prouty for his example helicopter is given in a later section. Because the output agreed well with known quantities, this method was applied to the forward flight regime. [Ref. 9]

IV. FORWARD FLIGHT CONDITION

A. EXISTING CODE

The forward flight regime was initially modified in exactly the same manner as the hover regime. The same input screens were used, but a logical branch was set in the program to access the forward flight subroutines if the airspeed entered was above 40 knots. The stability derivatives were evaluated in the same manner by varying about the nominal trim condition. However, the trim condition was perturbed by making small variations to the airspeed as opposed to the gross weight, as in the hover case. The A and B matrices were again constructed.

B. PROBLEMS ENCOUNTERED

When the A and B matrices were compared to known values of the example helicopter given by Prouty [Ref. 9], it was determined that the two were inconsistent. Therefore an extensive troubleshooting effort was launched. The code was run repeatedly to determine if some programming error had been made with the addition of the GUI screens. Wirth's thesis [Ref. 3] was used to verify the values obtained in the new hover regime, but an example for the forward flight case was not given.

C. IDENTIFING THE SOURCE OF THE ERROR

The main rotor derivatives and tail rotor derivatives are calculated using the formulas provided by Prouty. All of these formulas are functions of the rotor derivatives in forward flight obtained by Prouty through charts given in his book. These terms

involve the partial derivatives of C_T/σ, C_H/σ, C_Q/σ,a_{1s}, b_{1s} with respect to μ ,θ_0, and λ'. [Ref. 9]

These charts are not available to JANRAD and the correct values for these partial derivatives can not be obtained simply by varying the airspeed about the trim condition. Each time the aircraft is perturbed by varying the airspeed, μ is directly effected as an input to the program. However, the new trim solution produces new values for θ_0 and λ', thereby producing some kind of convoluted values for the derivatives of C_T/σ, C_H/σ, C_Q/σ, a_{1s}, b_{1s}.

A method was devised to vary each of the three parameters (μ ,θ_0, and λ') separately while holding the other two constant. In this way, the fifteen specific partial derivatives could be obtained.

D. IN SEARCH OF THE SOLUTION

The first attempt was to find a closed form solution for each of these derivatives. A NACA Technical Note written by Kenneth B. Amer and F. B. Gustafson entitled "Charts for Estimation of Longitudinal Stability Derivatives for a Helicopter Rotor in Forward Flight" [Ref. 10] was found. While some equations were given for the necessary stability derivatives, a chart was still needed to determine specific values for several of the variables. This was determined to be no better than Prouty's [Ref. 9] method of using the charts in his book directly to obtain the necessary values. Thought was given to converting these charts to tables and then having the program do a table look up to

determine the values, but it was determined that this would be far too time consuming and too specified to be of any real value.

E. THE SOLUTION

Finally it was determined that the appropriate method for obtaining these stability derivatives was to vary each parameter independently. While this may seem like an obvious choice, it proved to be a complicated one, because the many assumptions that had to be made. Many of the values used by the program are not explicit functions of each parameter, and therefore a determination had to be made for each variable in the program on the effect of changing these parameters independently.

V. PROUTY'S CHART METHOD FOR DETERMINING FORWARD FLIGHT STABILITY DERIVATIVES [REF. 9]

A. THE THEORY

The following rotor derivatives are needed to compute the stability derivatives that define the linear state space model of the helicopter (see Prouty's book [Ref. 9] for and explanation of each variable):

$$\frac{\partial C_T/\sigma}{\partial \mu}, \frac{\partial C_H/\sigma}{\partial \mu}, \frac{\partial C_Q/\sigma}{\partial \mu}, \frac{\partial a1s}{\partial \mu}, \frac{\partial b1s}{\partial \mu}$$

$$\frac{\partial C_T/\sigma}{\partial \theta_0}, \frac{\partial C_H/\sigma}{\partial \theta_0}, \frac{\partial C_Q/\sigma}{\partial \theta_0}, \frac{\partial a1s}{\partial \theta_0}, \frac{\partial b1s}{\partial \theta_0}$$

$$\frac{\partial C_T/\sigma}{\partial \lambda'}, \frac{\partial C_H/\sigma}{\partial \lambda'}, \frac{\partial C_Q/\sigma}{\partial \lambda'}, \frac{\partial a1s}{\partial \lambda'}, \frac{\partial b1s}{\partial \lambda'}$$

To determine the values for these derivatives, Prouty uses what is known as the graphical method, utilizing the Rotor Performance charts given in Chapter III of his book. These charts where originally developed by M.C. Cheney at Lockheed. The charts were generated using a computer program that used the equations described in Chapter III of Prouty's book. The charts, while based on a series of arbitrarily selected rotor parameters, are reported to be flexible enough to be used for rotors of various parameters. [Ref. 9]

By entering the charts at the trim values and varying each parameter independently, the rotor derivatives can be obtained. The following is a detailed

description of the process. Figure 2 is a graphical representation of the chart method for determining the stability derivatives as given in Prouty's book. [Ref. 9]

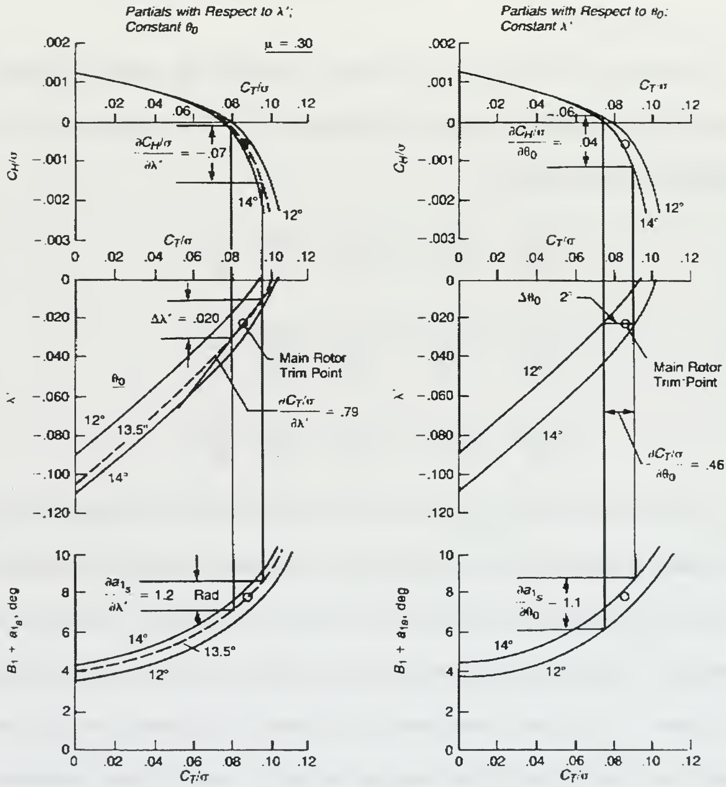

Figure 2. Illustration of Rotor Derivative Extraction from Performance Charts [Ref. 9]

B. STEP BY STEP DETERMINATION OF ROTOR DERIVATIVES

The values of the parameters at the trim condition must first be determined. Prouty [Ref. 9] gives the following values:

$$\mu = .30 \qquad \theta_0 = 13.5^\circ \qquad \lambda' = 0.23$$

μ is defined as the tip speed ratio, θ_0 is the trim collective pitch angle and λ' is the inflow parameter with respect to the tip path plane. To determine the value for the partial derivatives with respect to μ use the charts for μ=0.25 and μ=0.35. These values correspond to small perturbations around the trim value of μ. The charts are plotted against C_T/σ, so this is used as the intermediate variable.

Step one is to enter the chart of λ' vs. C_T/σ for μ=0.25. Find and record the value of C_T/σ for the trim values of λ' and θ_o. Step two is to enter the chart of C_Q/σ vs. C_T/σ. Find and record the value for C_Q/σ for the just determined value of C_T/σ and the trim value θ_0. Do the same for C_H/σ, a_{1s} and b_{1s}, always using the trim value for θ_o and the determined value for C_T/σ. Repeat this procedure using the μ=.35 charts.

To determine the values for the partial derivatives, simply take the difference between the determined values for the variables and divide by the difference of the parameters. For example:

$$\frac{\partial C_T / \sigma}{\partial \mu} = \frac{\frac{C_T}{\sigma}(\mu = 0.35) - \frac{C_T}{\sigma}(\mu = 0.25)}{0.35 - 0.25}$$

The partial derivatives with respect to θ_0 and λ' can be determined in the same manner. Using this method all fifteen of the rotor derivatives can be determined

C. PUTTING IT ALL TOGETHER

Once all of the non-dimensional rotor derivatives have been obtained, the formulas given in Prouty's book [Ref. 9] are used to determine the Main Rotor Derivatives. These are then combined with the tail rotor, horizontal stabilizer, vertical stabilizer, and fuselage derivatives to obtain the Total Forward Flight Derivatives. These are then used in the helicopter equations of motion to construct the linear model.

VI. JANRAD'S METHOD FOR DETERMINING FORWARD FLIGHT MAIN ROTOR DERIVATIVES

A. THE THEORY

JANRAD's approach is to perturb each of the parameters, μ, θ_0, λ', independently and compute the values for the variables C_T, C_H, C_Q, a_{1s}, and b_{1s}. This is exactly the same method that Prouty [Ref. 9] uses, however, instead of using charts, JANRAD computes the values for each variable specifically. A separate subroutine for varying each parameter was written due to the differing assumptions that had to be made for each case. Each subroutine both increases and decreases its parameter from the trim value and computes a value for each variable. A vector for the values of the parameters and a vector for each of the variables is created.

These vectors are passed to a subroutine that creates a first order curve and then computes the derivative of the curve. The slope of the curve at the trim value is then computed and this is used as the rotor derivative.

The procedure is simplistic, however many assumptions had to be made as to which variables in the trim routine had to be held constant and which had to be allowed to vary. The following section discusses the assumptions made and the criteria for various variables in the trim routine.

B. THE COMPUTATION

The general structure of each subroutine is the same and is modeled after the Trim.m subroutines written for the Performance Module. The entire subroutine is

written as a loop that is run three times, once for a value for the parameter below the trim value, once for the trim value of the parameter and once for a value above. The necessary aerodynamic terms are computed, the required lift, tip path plane angle, required thrust, coning angle, etc. The Tmcalc.m, Thrcalc.m, and Dmcalc.m subroutines written for the performance model are called once which gives the values for thrust, drag and torque of the rotor. C_T, C_H, C_Q, are then computed directly using there respective defining equations. Each of these M-files are run only once rather than several times in an iterative process as in the trim case for the Performance Module. A detailed description of how these M-files are structured can be found in Nicholson's thesis [Ref. 1]. The values for a_{1s} and b_{1s} are computed directly from explicit formulas. The following assumptions were made in each subroutine.

1. Tip speed ratio μ

Stab_calc_mu.m is the name of the M-file that calculates the variables while varying the parameter μ. Varying the airspeed by 15% in either direction, effectively varies the parameter μ. No assumptions have to be made, and therefore the subroutine runs very similarly to the trim routine, except that it is only run once and no adjustment to the collective pitch is made.

2. Collective Pitch θ_0

Stab_calc_to.m is the name of the M-file that calculates the variables while varying the parameter θ_0. θ_0 is varied by one degree in either direction. The

calculation for the coning angle is computed as a function of the change in θ_0 by the following equation:

$$\beta = \beta_0 - (\Delta\theta * (\tfrac{\beta_0}{\theta_0}))$$

This method for determining the coning angle is necessary because the original Trim.m routine varies β as a function of μ. If μ is held constant then β would be held constant and erroneous values for the parameters would be obtained. By varying the coning angle as a function of collective pitch change allows for the proper determination of the variables.

3. Inflow ratio with respect to tip path plane λ'

Stab_calc_la.m is the name of the M-file that calculates the variables while varying the parameter λ'. The inflow ratio was varied by 50% in either direction. The tip path plane was held constant. The coning angle was varied according to the following equation:

$$\beta = \beta_0 * (1 - \frac{\Delta\phi}{\theta_0 - \phi_0})$$

β again had to be varied as a function of inflow angle which is a function of λ' for the same reasons as stated above.

C. TAIL ROTOR DERIVATIVES

Due to the small variations in the derivatives for the tail rotor, and it's overall contribution to the total stability derivatives JANRAD does not compute tail rotor

derivatives in the same manner as the main rotor derivatives. While Prouty [Ref. 9] alludes that the chart method can be used to determine the tail rotor derivatives, JANRAD simply uses closed formed solutions for these parameters.

D. FUESELAGE DERIVATIVES

The fuselage derivatives are hard wired into the program using the charts given for the Prouty Example Helicopter in Appendix A of his book. These should be acceptable for most normally configured helicopter designs. If however it is determined that these values differ greatly from the Prouty curves, then the Cbodygrp.m M-file must be modified, to reflect the appropriate values. [Ref. 9]

E. THE REST OF THE COMPUTATIONS

A detailed description of the rest of the Stability and Control Module is explained in detail in Walter M. Wirth Jr.'s thesis [Ref. 3]. The computations differ only in the areas mentioned previously.

F. THE RESULTS

Once the stability derivatives are determined, JANRAD uses the same equations from Prouty's book to determine the Total Forward Flight Derivatives and to construct the linear mode. A detailed analysis of the results of JANRAD as compared to Prouty's for his example helicopter are given in a later section. [Ref. 9]

VII. OUTPUT

A. THE LINEAR MODEL SCREEN

The first screen that appears after the calculations are complete allows the user to see the actual A and B matrices and also indicates the vector for the states, and the inputs, Figure 3.

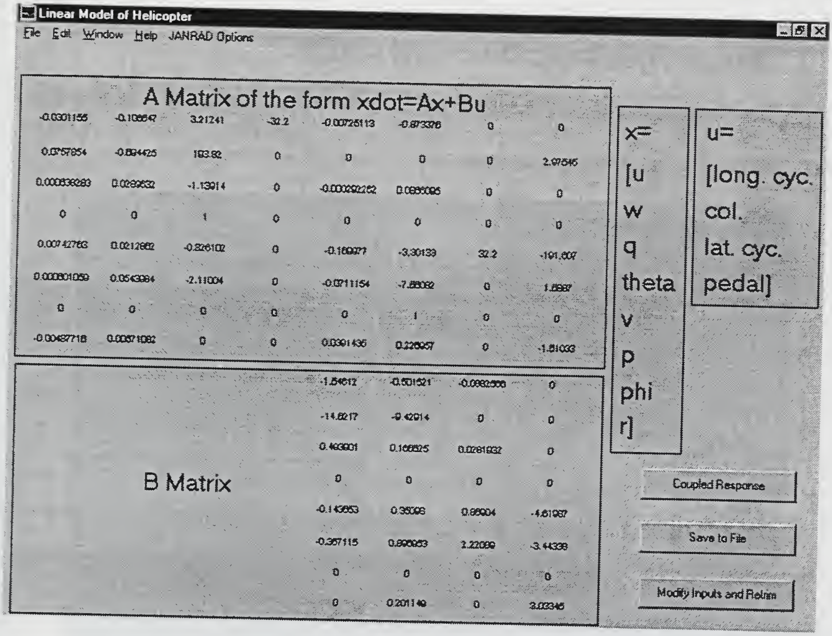

Figure 3. Linear Model of Helicopter Window

From this point the user can decide to save his output to a file, change input parameters an re-run the calculations, or proceed the time and frequency response screen.

The A matrix is an 8 X 8 matrix containing constants that completely describes the linear approximations to the equations of motion of the aircraft. The B matrix is a 4 X 8 matrix that describes the relationship between the input from the pilot controls and the input to the modeled system. With this B matrix a collective, cyclic, or directional input can be specified to the pilot controls and the appropriate rigging compensation are automatically made.

B. TIME AND FREQUENCY RESPONSE SCREEN

The Time and Frequency Response screen allows the user to analyze the response of the helicopter to specified inputs, Figure 4.

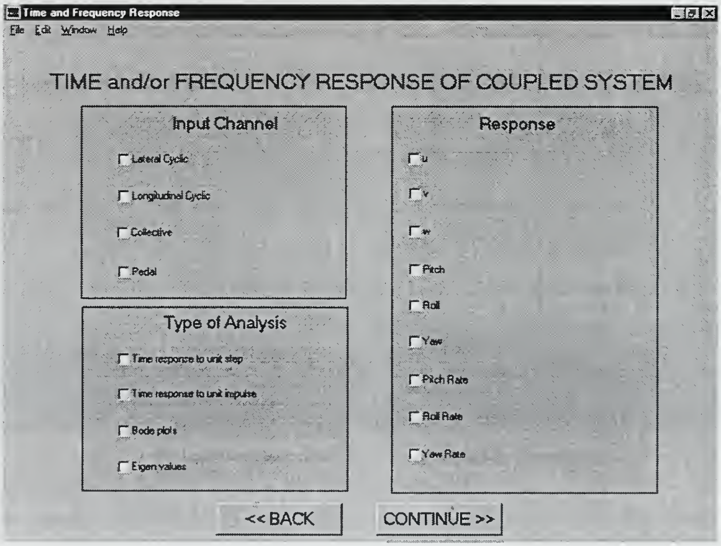

Figure 4. Time and Frequency Response Window

Step or impulse inputs can be simulated in either the collective, cyclic, or directional channels and responses of any of the eight states can be observed. Figure 5 is an example of one of the output plots generated for the response of role rate to a lateral cyclic step input.

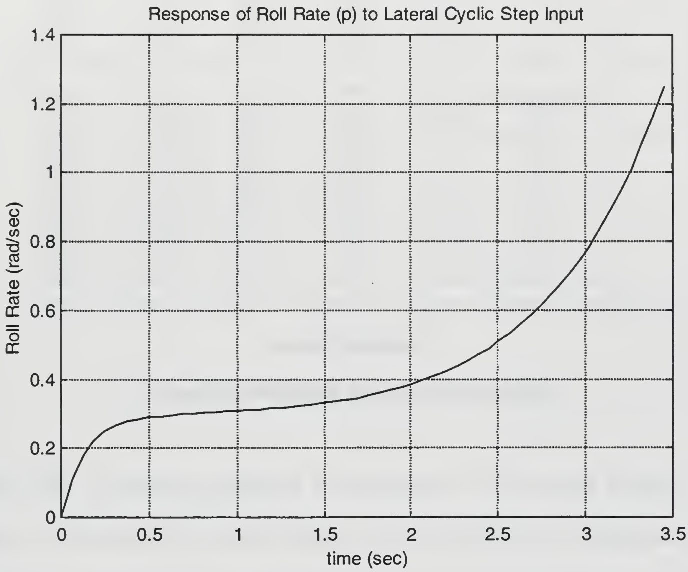

Figure 5. Example of Step Response Output Plot

A frequency analysis can also be obtained by specifying Bode plots. In this manner, the Bode plots for each channel can be generated. Figure 6 is an example of a Bode plot for the system between roll rate an lateral cyclic input

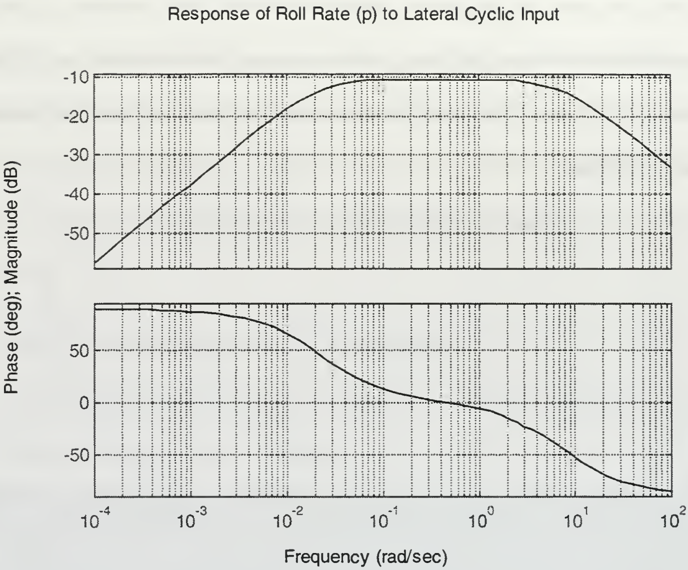

Figure 6. Example of Bode Plot Output

A stability analysis can be performed by specifying eigenvalues. This option generates and plots the eigenvalues of the A matrix which are the poles of the system. The coupled eigenvalues along with the decoupled lateral and longitudinal eigenvalues can be viewed in graphical form. Figure 7 is an example of a plot of the coupled eigenvalues. If the numerical values are desired, or a more detailed stability analysis is needed, the user can load the output file containing the A and B matrices at the command line or create a unique M-file to generated specific analyses.

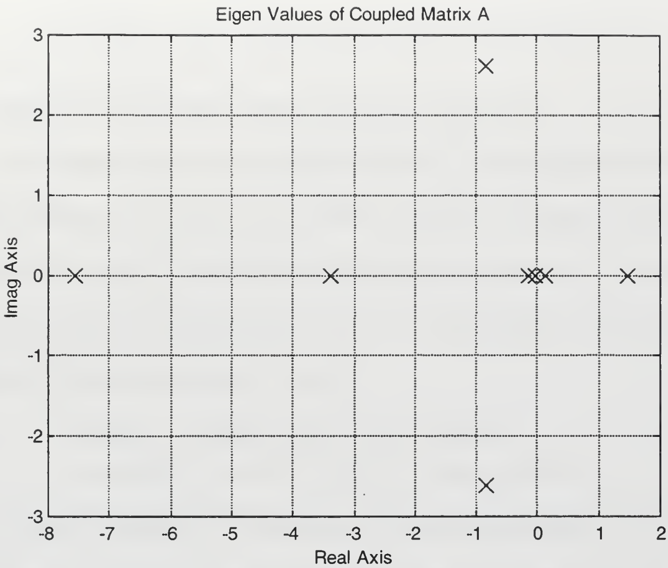

Figure 7. Example of Eigenvalue Output Plot

VIII. COMPARISON WITH KNOWN HELICOPTER MODELS

A. PROUTY'S EXAMPLE HELICOPTER IN A HOVER

Prouty [Ref. 9] gives the characteristic equation of the longitudinal motion for his example helicopter in his book. An extensive analysis of the lateral motion of the helicopter in this regime is not performed, therefore the comparison for the hover mode will be done only in the longitudinal direction. The following are the characteristic equations for the example helicopter in a hover.

$$\text{Prouty}: \; s^4 + 1.02s^3 + 0.21s^2 + 0.12s + .034 = 0$$
$$\text{JANRAD}: \; s^4 + 1.113s^3 + 0.2279s^2 + 0.1383s + .0374 = 0$$

While the coefficients of the polynomial to not match exactly they are very close. A more intuitive method for comparing the two systems is to observe the eigenvalues. Figure 8 plots the roots of the above equations, which are the eigenvalues of each system as given below:

Prouty: $s_1 = -0.28$, $s_2 = .076+0.360$, $s_3 = 0.076 - 0.360i$, $s_4 = -0.89$

JANRAD: $s_1 = -0.2704$, $s_2 = .0713+0.3679I$, $s_3 = 0.0713 - 0.3679I$, $s_4 = -0.9851$

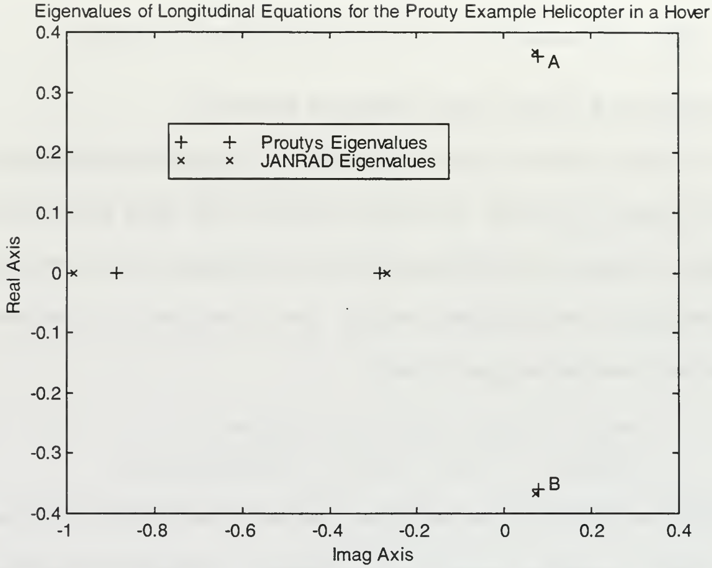

Figure 8. Eigenvalues of Longitudinal Equations in a Hover

A quick glance at the plot will tell you how well JANRAD is performing. The eigenvalues for the two systems are very close. The dominant poles (A and B) in each case lie very close to each other, providing similar response in the time domain.

B. PROUTY'S EXAMPLE HELICOPTER IN FORWARD FLIGHT

The analysis for the forward flight regime is broken into the longitudinal and lateral cases. While a coupled response analysis can be done, decoupling the equations will emphasize the similarities and differences between Prouty's [Ref. 9] numbers and JANRAD's and will also allow for easier determination of the effected channels. The

results from the forward flight mode do not compare as well as for the hover regime. This is due to the difficulty of the determination of the rotor derivatives. However, as this program is to be used only for preliminary design, it has been determined that the numbers correspond well for this purpose.

1. Lateral Motion

The following are the characteristic equations for the lateral motion of the example helicopter at 115 knots.

$$\text{Prouty: } s^4 + 8.460s^3 - 17.68s^2 + 45.54s + 2.2548 = 0$$
$$\text{JANRAD: } s^4 + 9.3411s^3 + 20.6391s^2 + 57.7401s + 1.6851 = 0$$

Figure 9 plots the roots of the above equations, which are the eigenvalues of each system as given below:

Prouty: $s_1 = -6.602$, $s_2 = -0.7841+2.4317i$, $s_3 = -0.7841-2.4317i$, $s_4 = -0.05058$

JANRAD: $s_1 = -7.6237$, s_2 $-0.8441+2.6044i$, $s_3 = -0.8441-2.6044i$, $s_4 = -0.0295$

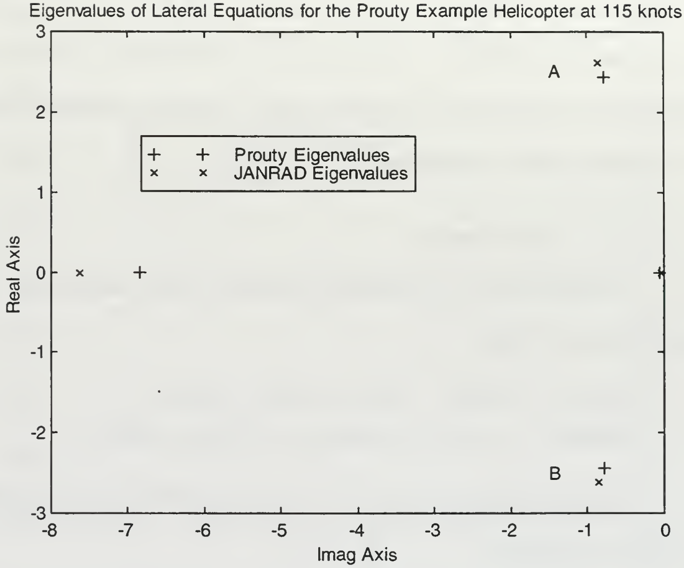

Figure 9. Eigenvalues for Lateral Motion in Forward Flight

This plot will again tell you how well JANRAD is performing. The dominant poles (A and B) in each case lie very close to each other, providing similar response in the time domain. Figure 10 plots a time history for the response of yaw rate to an impulse pedal input using the uncoupled lateral equations. It is evident that the two systems have very similar time histories.

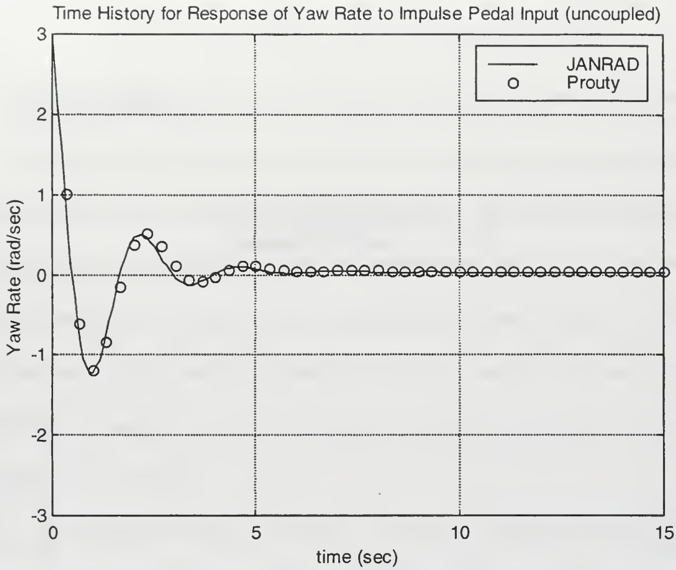

Figure 10. Time History Comparison

2. Longitudinal Motion

The following are the characteristic equations for the longitudinal motion of the example helicopter at 115 knots:

$$\text{Prouty}: \; s^4 + 1.545s^3 - 2.61s^2 + 0.0288s + 0.0949 = 0$$
$$\text{JANRAD}: \; s^4 + 1.8637s^3 - 4.7614s^2 + 0.1108s + 0.08500 = 0$$

Figure 11 plots the roots of the above equations, which are the eigenvalues of each system as given below:

Prouty: $s_1 = -2.907$, $s_2 = -0.1710$, $s_3 = 0.1828$, $s_4 = 1.085$

JANRAD: $s_1 = -3.2958$, $s_2 = -0.1418$, $s_3 = 0.1255$, $s_4 = 1.4484$

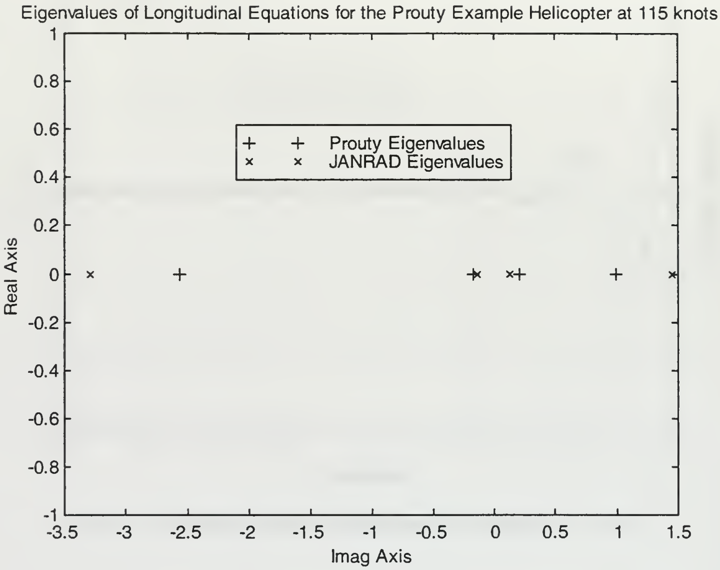

Figure 11. Eigenvalues for Longitudinal Motion in Forward Flight

C. HANDLING QUALITIES

The charts in Appendix B show how the handling qualities of the Prouty example helicopter [Ref. 9] using the model given in his book compare to that computed using JANRAD. It graphically depicts the validity of the JANRAD program.

IX. CONCLUSIONS AND RECOMENDATIONS

A. CONCLUSIONS

The ability to produce a linear model of a helicopter for stability and control analysis is an invaluable tool in the design process. With the development of the JANRAD Stability and Control Module, the generation of this linear model has become as easy as entering a few parameters and pressing a few buttons. The user can modify any one of the various inputs and re-compute the model in a matter of minutes. In this way, many different configurations can be obtained and the one best suited for the particular application can be selected.

It has been shown that the solutions produced by JANRAD match closely with those presented by Prouty in his book [Ref. 9]. While the techniques used to compute the stability derivatives are very similar, the method for computing the terms comprising these values differs greatly. This fact only lends to further trust that the values produced by both Prouty [Ref. 9] and JANRAD are correct.

The output screens created enable the user to quickly and easily evaluate the response of his design. Again, changes can quickly be made and the analysis repeated to ensure the best possible design.

With the evolution of the Stability and Control Module of JANRAD the helicopter designer can concentrate on design rather that spending time devising a means of analysis.

B. RECOMMENDATIONS

Due to the time limitations and the necessity to redesign the method for computation of the stability derivatives the Rotor Dynamics Module of JANRAD was not developed. The development of this module and the addition of it to JANRAD would greatly enhance the design process. It can easily be added to JANRAD from the Options screen.

While the Performance module originally was designed for fan in tail and NOTAR capabilities, somewhere in the design process this function has been abandoned. Not only does the Stability and Control module not support it, but the Performance Module has also lost this capability. A warning screen has been created to prevent the user from attempting this type of design.

Finally, due to the lack of another valid linear model of a different helicopter, only Prouty's example helicopter was used to verify the output. In order to ensure that the output is valid for a variety of configurations, the program should be compared with other known linear models.

APPENDIX A. USER'S GUIDE

The JANRAD version 6.0 Users Guide is written as a brief introduction to the Joint Army/Navy Rotorcraft Analysis and Design computer program. It is intended to explain the basic features and operation of the program and assumes a basic knowledge of helicopter mechanics and the use of the MATLAB® programming language by The MathWorks® Inc.

A. SYSTEM REQUIRMENTS

JANRAD version 6.0 requires MATLAB® version 5.0 or MATLAB Student Edition version 5.0 or higher. It will not run on any previous versions. JANRAD version 6.0 will fit on a single 1.44 MB floppy disk and will need that much memory available for installation. JANRAD requires only the hardware to support MATLAB® 6.

B. INSTALLATION

The recommended installation of JANRAD version 6.0 is accomplished by first creating a subdirectory of MATLAB called JANRAD. The entire contents of the JANRAD version 6.0 floppy disk should be copied into this directory. Include all M-files and .mat files. JANRAD version 6.0 will not run without all of the .mat files.

It is recommended that this new subdirectory be added to the MATLAB 5 search path. This procedure will eliminate the need to change the working directory from the command line each time JANRAD version 6.0 is run and allows you to work from a

floppy disk if desired. Adding the subdirectory to the search path is accomplished by selecting *File*, *Set Path...* from the File menu. Change the current directory to the new Janrad98 subdirectory by using the *Browse* button. Then press the *Add to Path* push button. You will then be given the option to save the new path or just use the new path for the active session. It is recommended to save this path. Figure A.1 shows the MATLAB® Path window.

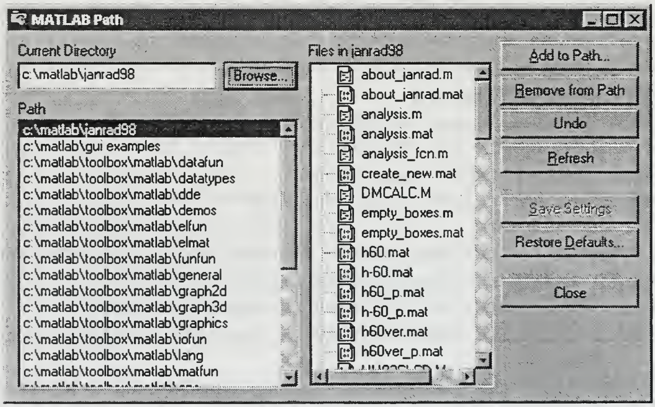

Figure A. 1. MATLAB 5 Path Window

C. STARTING JANRAD VERSION 6.0

Typing janrad98 (lowercase, one word) at the command line prompt of a current MATLAB session starts JANRAD version 6.0. This action will launch the JANRAD version 6.0 welcome window shown in Figure A.2.

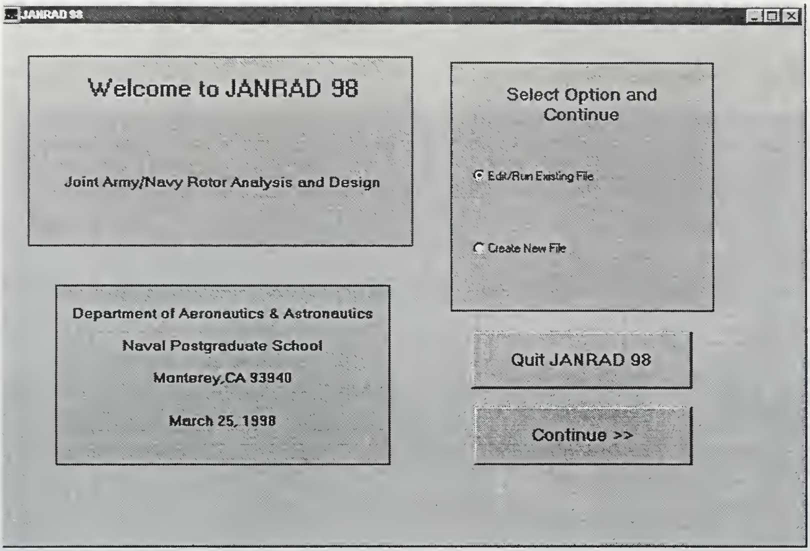

Figure A. 2. JANRAD version 6.0 Start Up Window

D. PERFORMANCE MODULE

As an example, the use of this program will be demonstrated by selecting a previously saved input data file and changing the weight, airspeed and pressure altitude. User defined blade elements and blade twist will be entered. Tail rotor parameters will be verified but not changed. The input and output files will be saved and printed. After the performance analysis is complete, we will then iterate on airspeed from 80 to 100 knots in increments of 5 knots.

First, from Figure A.2, select the Edit/Run Existing File radio button. It is usually easier to edit an existing file because Create New File will not give you the chance to

change the working directory if desired. Once the selection has been made, press the *Continue* >> button.

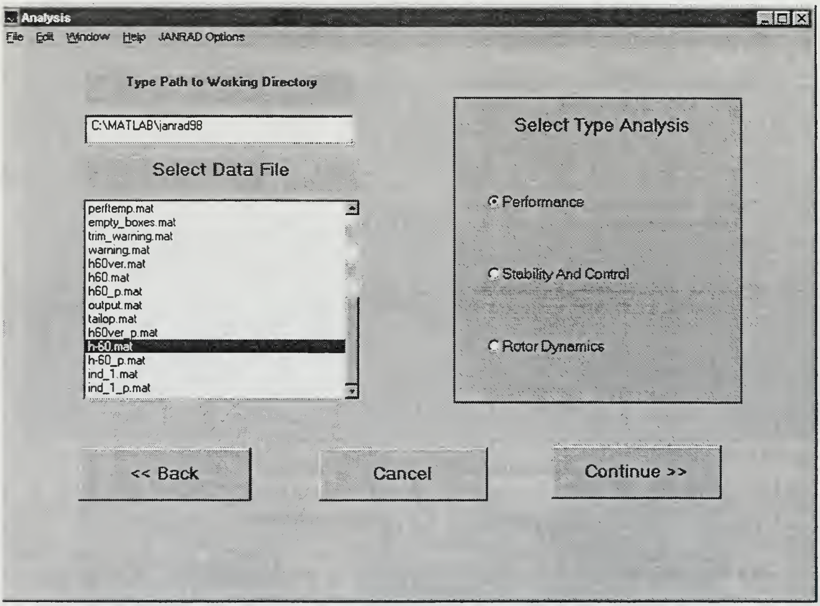

Figure A. 3. Selecting a File to Edit

The next figure window to appear is shown in Figure A.3. This window allows you to change the working directory and select an input data file. To change the directory, type or edit the desired path in the edit box. All of the .mat files listed in the working directory are displayed in the list box. Input data files are saved as *filename.mat*. A note of caution here, each GUI window also has an associated *guifilename.mat* file. The user should name input/output data files using helicopter aircraft designations such as

UH_60A, h_99 or h _design1 to differentiate from JANRAD version 6.0 GUI files. Next, select an input file to edit by clicking on the file name. For this example, we will edit the UH_60A.mat file. Then press *Continue* >>.

The Performance Input window will be displayed as in Figure A.4. The input data will be displayed within the appropriate edit boxes. Any or all of the parameters can be changed at this point without altering the original data file. You will have the opportunity to save the new data if you chose after the analysis has been completed.

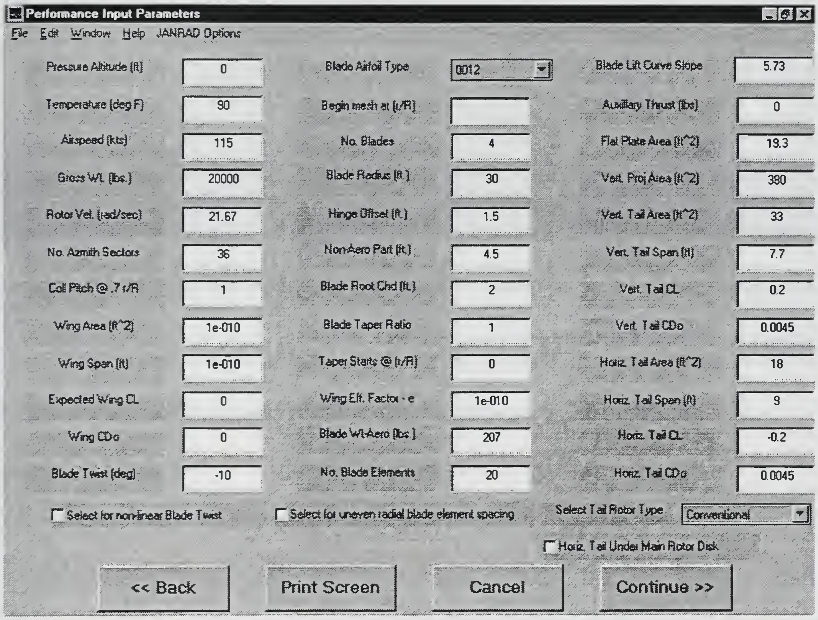

Figure A. 4. Performance Input Parameters

The airspeed, weight and pressure altitude can be edited by highlighting and typing 100, 17,000 and 2,000 in the respective edit boxes. Pressing the enter key is not

necessary to enter the new value. Using the Tab key or clicking on another edit box or control will enter any changes. If the user wishes to mesh airfoils, click the Blade Airfoil Type scroll bar, and select *Airfoil_Mesh.* The Begin Mesh at (r/R) box will be enabled and a value may be entered. At this point, select both the non linear blade twist and uneven blade element spacing blocks. Note that the blade twist and number of blade element boxes are disabled. The Print Screen button will print a draft copy of the GUI window with the displayed values if desired. It however, will not record the file name for which the values are stored. Now press the *Continue* >> button.

From the Performance Input window, JANRAD version 6.0 will call the Compound Helicopter and Tail Rotor Parameters window, Figure A.5. In our example we are dealing with a conventional helicopter, therefore no wing or auxiliary thrust is present. We do, however, need to verify the tail rotor parameters which have been either loaded from the data file or calculated. The parameters will be entered in the appropriate tail rotor type. Press *OK* when ready to continue.

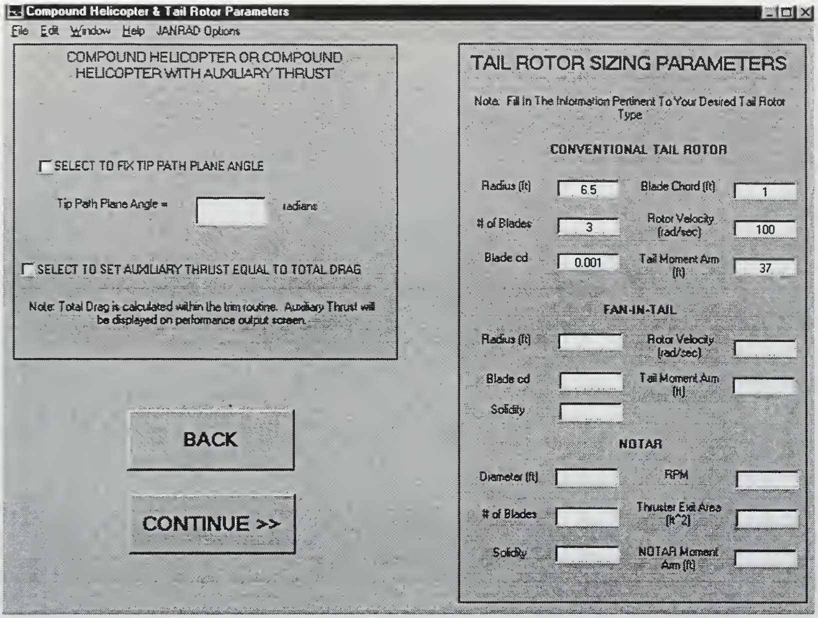

Figure A. 5. Enter Compound Helicopter and Tail Rotor Parameters

With either non-linear blade twist or uneven blade element spacing selected, JANRAD version 6.0 will next go to the Blade Element page, shown in Figure A.6.

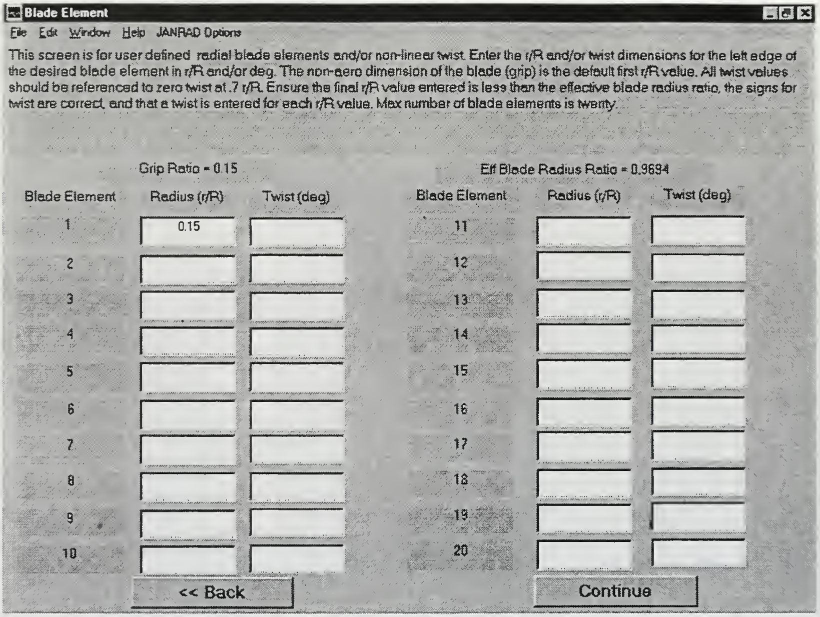

Figure A. 6. Enter Uneven Blade Elements & Non linear Blade Twist

The grip ratio contained in the loaded data file will automatically be displayed along with the effective blade radius. The user can enter up to twenty blade elements and the corresponding twists. **IMPORTANT!** Ensure dimensions are entered to the left edge of the blade element from the blade root. JANRAD version 6.0 will automatically calculate the values at the center of the blade elements. Also, do not enter any value greater than the effective blade radius or an error message will appear. After entering the desired values press *Continue.*

Next, JANRAD version 6.0 will call the Iteration Method window as shown in Figure A.7.

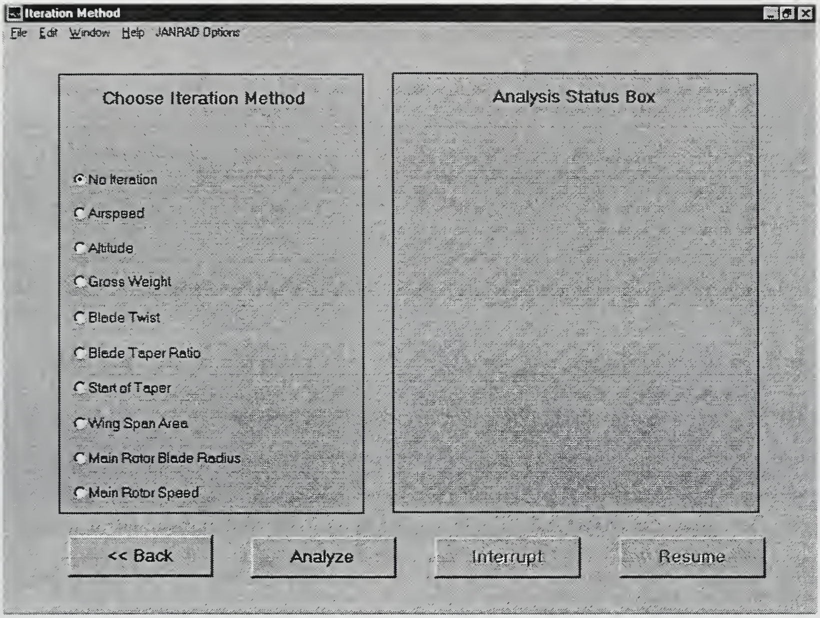

Figure A. 7. Iteration Method / Analysis Window.

JANRAD version 6.0 will call its computational routines from this window. By choosing *No Iteration* and *Analyze*, JANRAD version 6.0 will run the parameters selected from the previous window. The *Analyze* pushbutton initiates the computational routines. All controls on the GUI will be disabled except the Interrupt pushbutton. The Analysis Status Box will display the performance routine status, clock, iteration number, and iteration parameter value as JANRAD trims the rotor and adjusts the collective and cyclic

mathematically. The Interrupt button will halt the routine and enable the Resume control and JANRAD Options menu on the GUI. This will allow the user to change parameters, quit or return to beginning. The Resume button will continue with the performance routine where it originally interrupted. It is worth noting that the Interrupt button will not always respond immediately. However, once MATLAB finishes its current line evaluation, the calculation will pause.

The Status Box will inform the user when calculations are complete. The Performance Output window will be displayed automatically. The Performance Output window shown in Figure A.8 displays the performance results. These results can be saved and the screen printed from this window. However, it is recommended to print the saved input and output files through the next window. By pressing the *Options >>* push button, the saved input and output files can be printed simultaneously and in a more usable format. The input/output files can be saved after activating the checkboxes, typing a file name and pressing the *Save* or the *Options >>* push buttons.

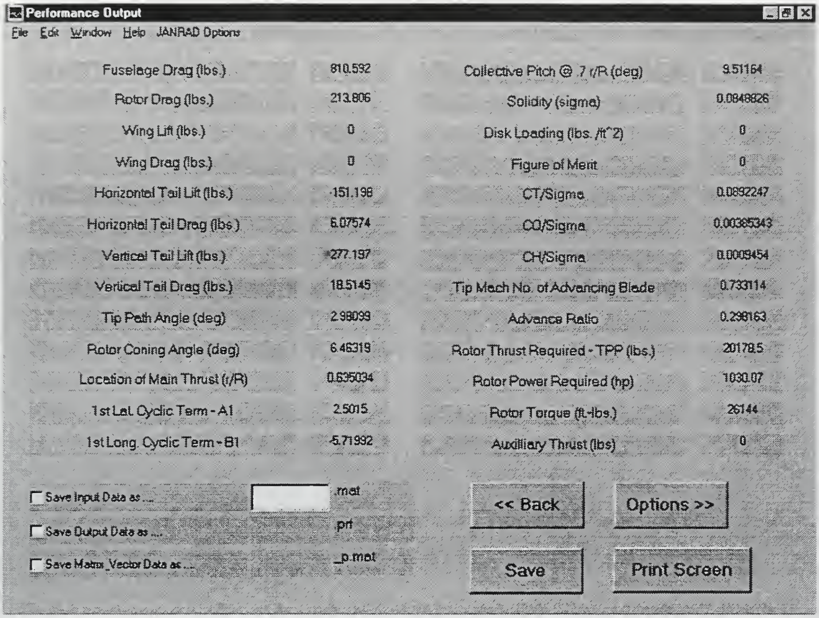

Figure A. 8. Performance Output Window

The Options window in Figure A.9 provides the capability to print the latest files, go to create plots screens, and eventually, select additional analysis routines. At this time however, the Stability and Control and Rotor Dynamics routines have not been completed. If selected, you will be reminded of this limitation.

From here, we will go back to calculate performance parameters by varying airspeed. This is done by pressing the *Change Iteration Method* radio button and the *Continue* >> push button. This will go back to Figure A.7. To vary airspeed, press the *Airspeed* radio button and then *Analyze*.

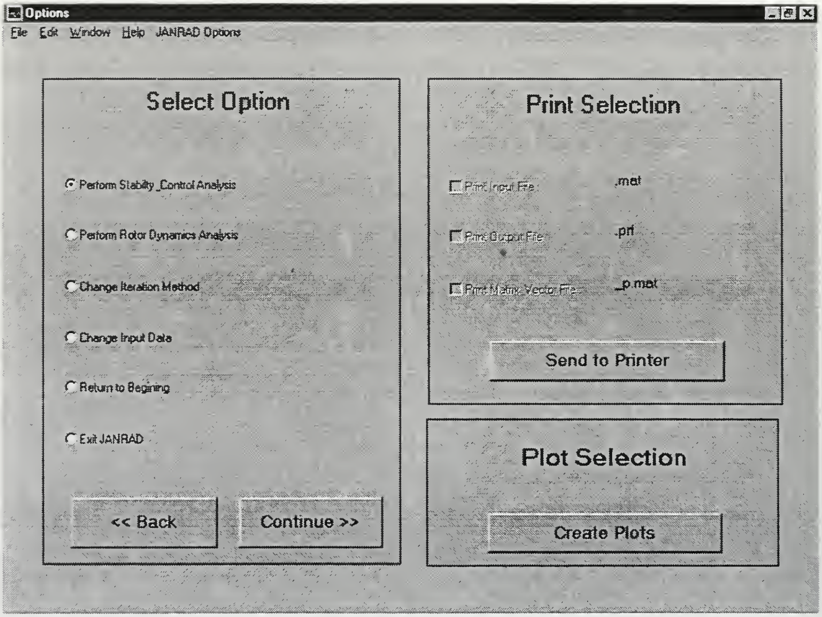

Figure A. 9. Options Window.

Figure A.10 shows the Iteration Parameters window. For example, by typing 80, 100 and 5 in the appropriate edit boxes and selecting *Analyze*, the performance routine will calculate various performance results with respect to airspeed varying from 80 to 100 knots. Later, the user will be able to create plots of many of these output parameters. Note: the *Aspect Ratio* edit box and *HIGE* check box are only enabled when *Altitude Iteration* or *Wing Span Area* is selected.

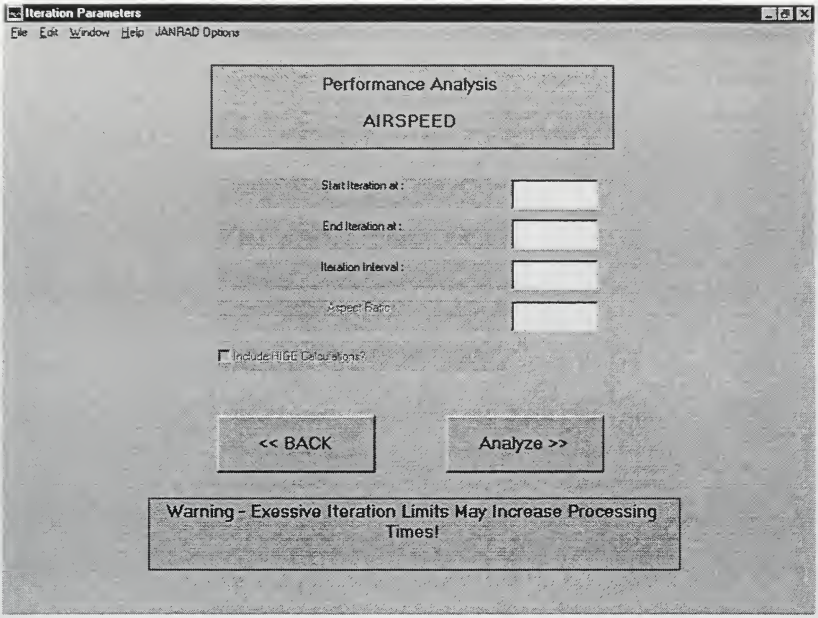

Figure A. 10. Iterations Parameter Window.

Figure A.11 shows the Create Plots screen for the Airspeed iteration method. Any or all of the plots may be selected. The plots are created and minimized as JANRAD automatically recalls the Options window, Figure A.9. Each iteration method has its own create plots screen. Some plots require additional user input prior to creating them.

Figure A.12 shows an example airspeed iteration subplot. These plots are primarily used to examine trade off studies during the design process.

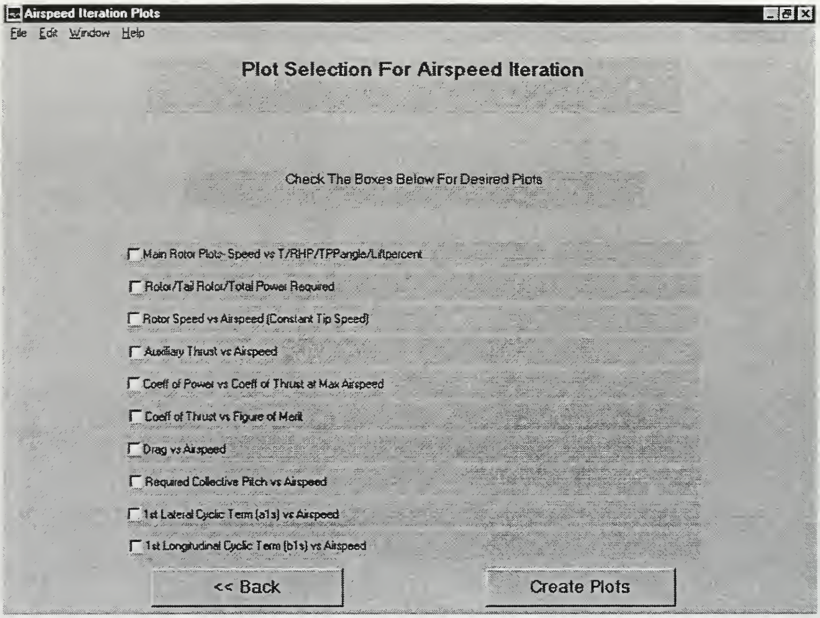

Figure A. 11. Example Iteration Method Create Plot Window

The M-file create_plots.m contains the code for all of the iteration method create plot screens. If any additional plots are desired, changes can be made to this file. However, the plots should be substituted for those plots already existing, and the existing code should be commented out, not removed. Adding plots requires changes to multiple files for proper operation and should not normally be attempted. Always document changes made to the code for future users.

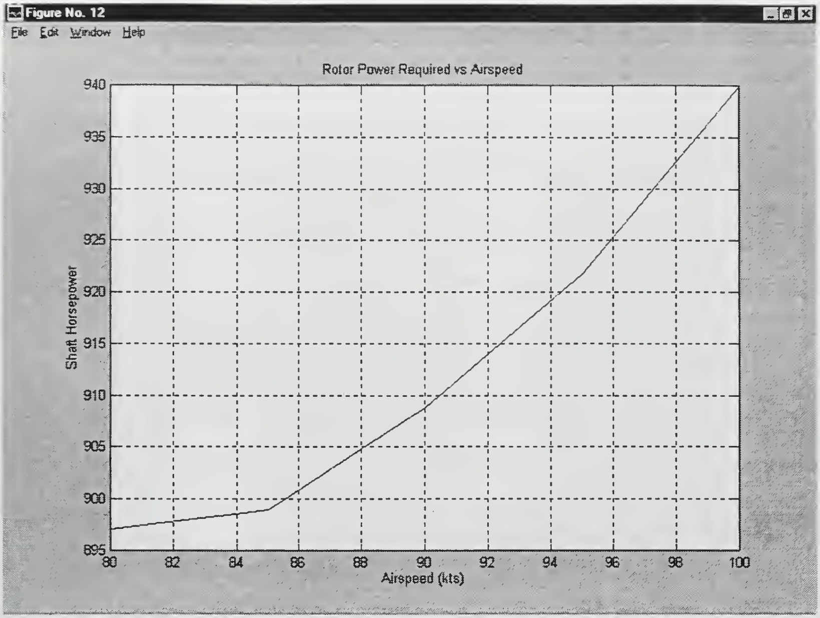

Figure A. 12. Example of Airspeed Iteration Output Plot.

E. HINTS FOR JANRAD VERSION 6.0 PERFORMANCE MODULE

The following paragraphs list some recommendations for the most efficient use of JANRAD VERSION 6.0. They are a guide based on observation, experience and knowledge of the code. Any other recommendations should be addressed to the Helicopter Design Instructor at the Naval Postgraduate School for implementation into the next version of JANRAD version 6.0.

JANRAD version 6.0 was designed for robust operations. However, because this is the first version to utilize a Graphical User Interface, not all cases of user inputs have been exercised. If the program appears not to be working properly, quit JANRAD using the options menu available on all but the first window. This action will quit JANRAD, close all MATLAB figure windows and clear the base workspace. Restart JANRAD by typing janrad98 at the command line.

F. STABILITY AND CONTROL MODULE

In order to run the stability and control portion of JANRAD the Performance Module must me run first. **WARNING: The analysis must be done in the No Iteration mode in order to ensure proper calculation.**

Figure A.13 and A.14 show the first screens to appear if the Stability and Control box is clicked on the Options Screen, Figure A.9. The current version JANRAD is not configured for NOTAR analysis, therefore zeros should be entered for values in that section. If a compound helicopter is being designed then the wing parameters are to be entered, if not then enter zeros for the wing values.

Rigging values can be difficult to evaluate. If they are not know then enter a value of 1.0 for these values. This will not effect the A matrix, however the B matrix will be effected. Entering a value of 1.0 will result in a one to one correlation of input from the controls to the input to the system.

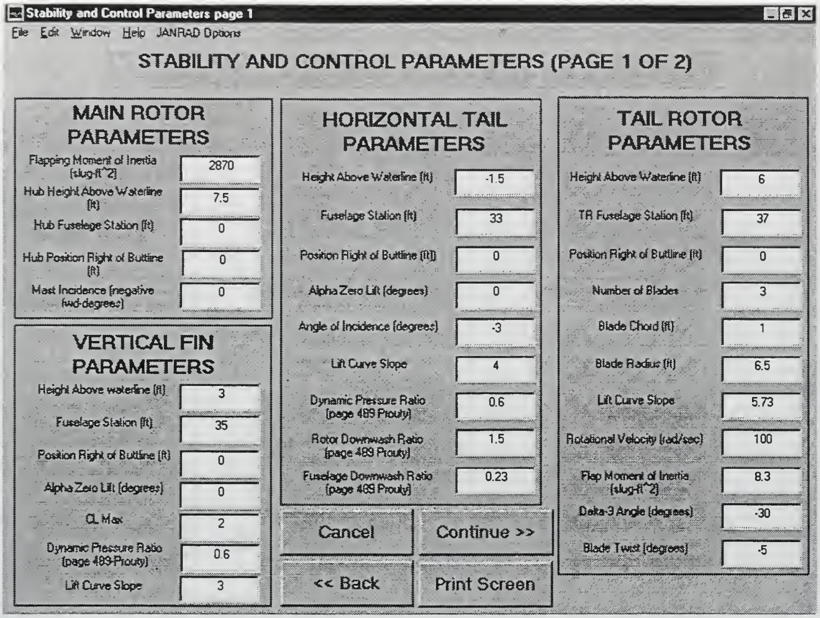

Figure A. 13. Stability and Control Parameter Page 1.

All of the required dimensions are listed in the figure and page numbers for Prouty's [Ref. 9] are given.

After all the parameters have been entered and the *Continue* >> button is pushed, the next screen is Stability and Control Status, Figure A.15. This screen allows the user to see the calculations in progress. It also allows the user to know that the calculations are still being made. If the run elapsed time has stopped changing then there is probably a

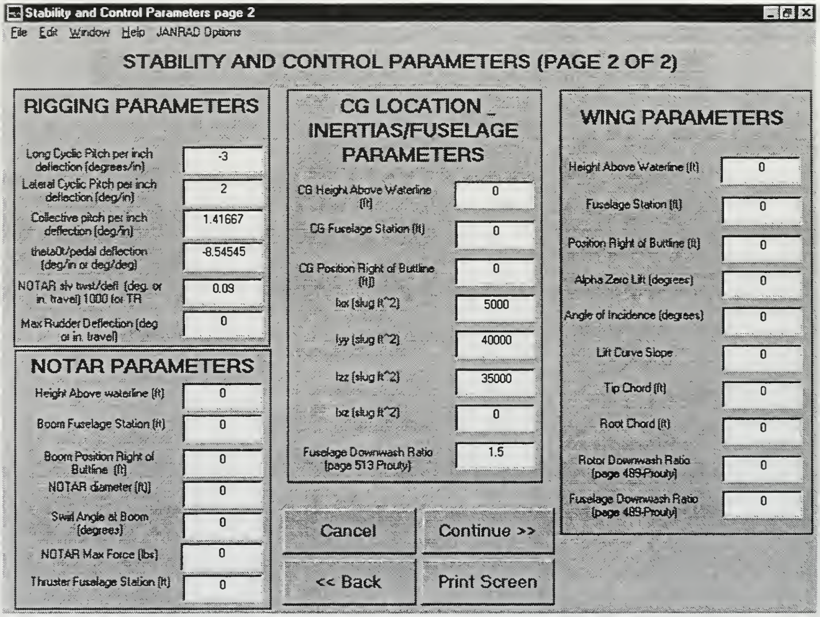

Figure A. 14. Stability and Control Parameters Page 2.

hang up in the program and the user should start over.

After the calculations are completed the linear model of the helicopter is displayed, Figure A.16. This screen also shows the order of the states and the inputs. From this screen the user has three options. Pressing the *Save to File* button brings up a screen prompting for a file name in which to save the input, and output values. The input values for both the Performance Module and the Stability and Control Module are saved to a single file. The output file will contain only the A and B matrices that comprise the

linear model. The output from the Performance Module can still be accessed from the file that was saved at the end of that module

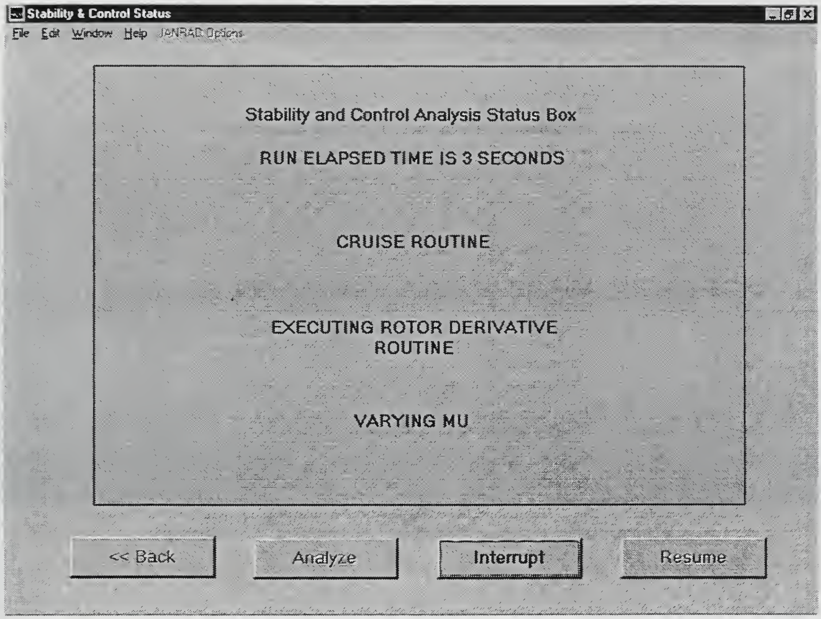

Figure A. 15. Stability and Control Status.

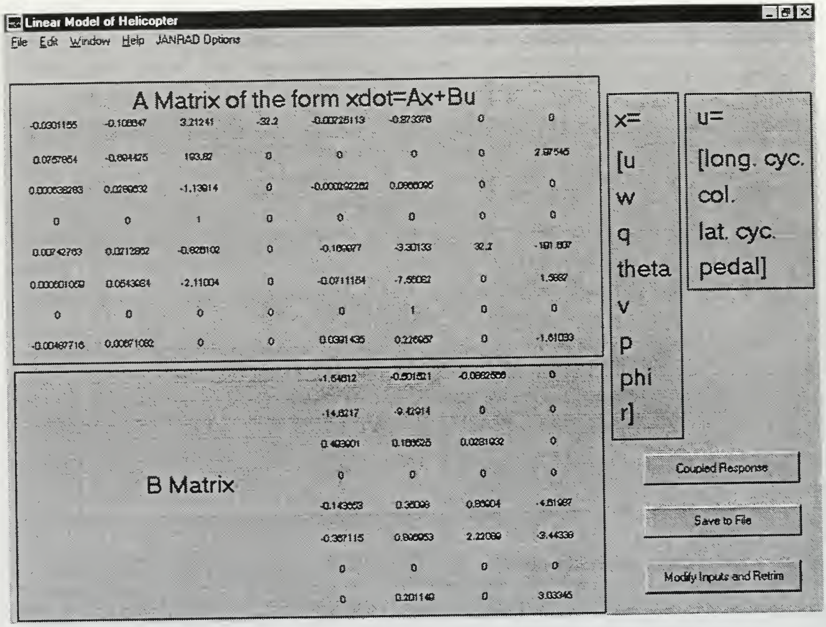

Figure A. 16. Linear Model of Helicopter Window

Pressing the Modify inputs and re-trim will send the user back to the input page for the Performance Module. If any inputs are changed then a new trim solution must be obtained before the Stability and Control Module can be run. The program is set up so that even if no inputs are changed in the Performance Module, the trim solution must still be re-computed. This was done to simplify the program and to ensure that a correct solution for the Stability and Control Module is obtained.

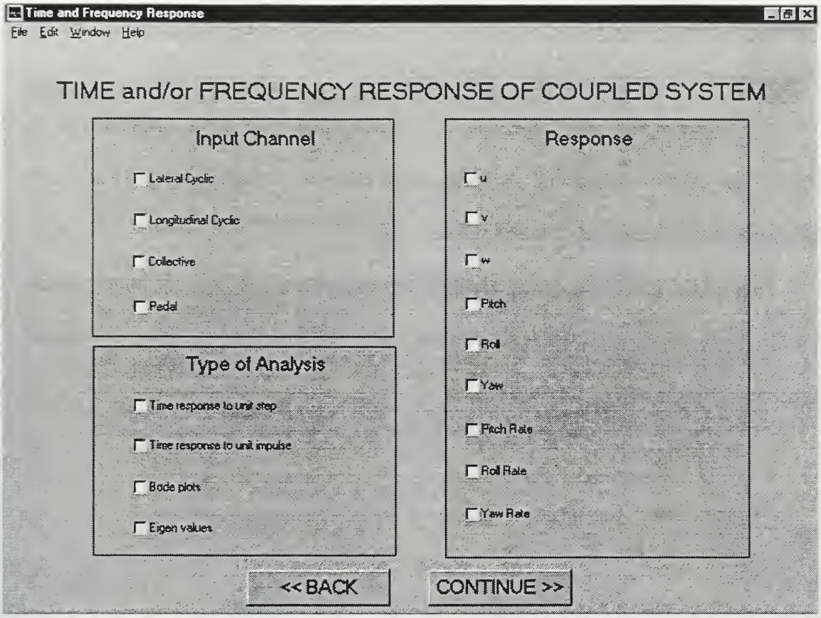

Figure A. 17. Time and Frequency Response Window

If the *Coupled Response* button is pressed, the screen shown in Figure A.17 will appear. This window will allow the user to perform various time and frequency analyses. The purpose of this screen is to give the user a idea of the general handling qualities of his design. If an extensive analysis is desired then the user should create his own MATLAB program and load the values for the A and B matrices save in the previous screen.

The user must select the desired input channel from the four choices in the left upper corner of the screen. Only one input can be selected at time. Once an input is

selected, the other three choices will be shaded and the user will not be able to select these until his previous selection had been deleted.

Next the user must select the response channel. One or all of the channels can be selected. The user is warned that selecting all the channels can cause a significant time delay depending on the speed of the hardware in use.

Finally, the user must select the type of analysis desired from the menu in the lower left hand corner of the screen. The user must then press the *Continue* >> button and the desired plots will be displayed.

APPENDIX B. HANDLING QUALITIES

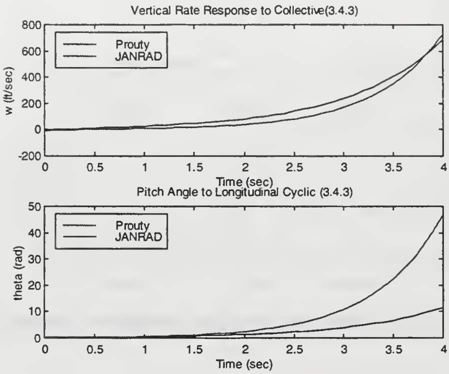

Figure B. 1. Flight Path Control

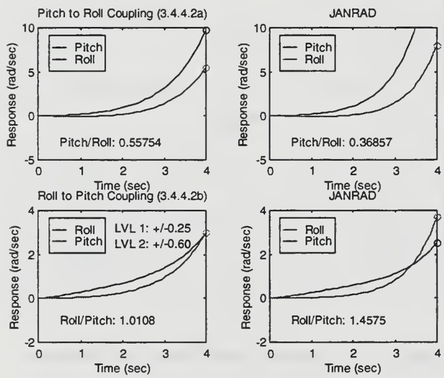

Figure B. 2. Inter Axis Coupling

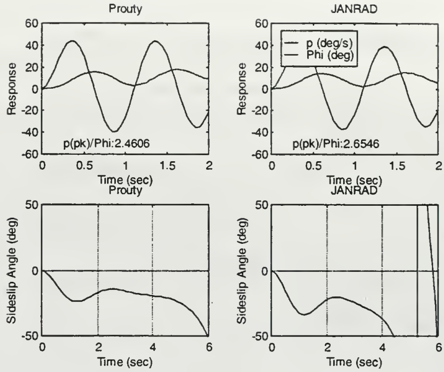

Figure B. 3. Attitude Quickness and Large amplitude Heading Changes

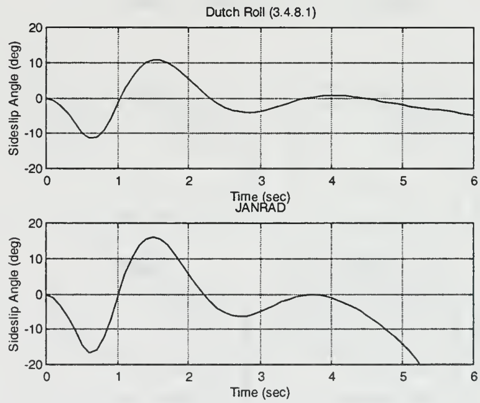

Figure B. 4. Lateral Directional Oscillations

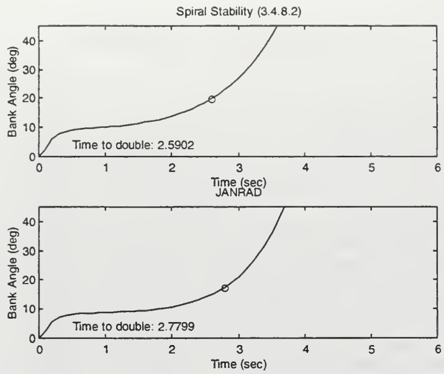

Figure B. 5. Spiral Stability

APPENDIX C. VARIABLE LIST

Computational Variables

a	lift curve slope of rotor system airfoil
A	rotor disk area
Ab	area of blades
Abt	area of tail rotor blades
Adisk	area of rotor disk
Afh	fuselage equivalent flat plate drag area
afoil	rotor system airfoil type (HH02/VR12)
Afv	vert. projected area (fuselage area under disk)
Ah	area of the horizontal stab
ah	lift curve slope of the horizontal stab
A1	lateral cyclic pitch
alph	angle of attack of horizontal stab
alpha	angle of attack, rotor blade radial segment
alphas	angle of attack WRT shaft axis
alphaT	rotor tip path plane angle
alploh	alpha zero lift of the horizontal stab
alplov	alpha zero lift of the vertical stab
alplow	alpha zero lift of the wing
alpv	angle of attack of vertical stab
alpw	angle of attack of wing
a1s	longitudinal flapping
altpp	mean alpha TPP
ao	main rotor coning angle
aot	tail rotor coning angle
Ap	projected area of fuselage under rotor
At	disc area of the tail rotor
at	tail rotor blade lift curve slope
Av	area of the vertical stab
av	lift curve slope of the vertical stab
Aw	area of the wing
aw	lift curve slope of wing
b	number of rotor blades
B	tip loss parameter
beta	trim sideslip angle
betao	rotor coning angle
betat	geometric angle, rotor blade radial segment
bh	spari of the horizontal stab
bhoriz	span, horizontal tail
B1	longitudinal cyclic pitch
B1s	lateral flapping
bv	semispan of the vertical stab
bvert	span, vertical tail
bw	span of the wing
bwing	span, wing
c	main rotor chord
cblade	chord, rotor blade
CD	drag coefficient, rotor blade radial segment
CDhoriz	drag coefficient, horizontal tail
cdoh	Cdo of the horizontal stab
CDohoriz	profile drag coefficient, horizontal tail
CDovert	profile drag coefficient, vertical tail

cdow	Cdo of the wing
CDowing	profile drag coefficient, wing
CDvert	drag coefficient, vertical tail
CDwing	drag coefficient, wing
CH	rotor H force coefficient
ch	coefficient
CH_sig	CH/solidity
Chsig	Ch/sigma
CL	lift coefficient, rotor blade radial segment
CLhoriz	lift coefficient, horizontal tail
CLvert	lift coefficient, vertical tail
clvertmax	maximum cl of the vertical stab with full rudder
CLwing	lift coefficient, wing
cmun	design Cmu for NOTAR boom in hover (.3 .6)
CON_b	Conventional tail rotor # of blades
CON_cdo	Conventional tail rotor cd0
CON_lt	Conventional tail rotor moment arm
CON_omega	Conventional tail rotor speed
CON_R	Conventional tail rotor radius
ct	tail rotor chord
cpipitch	Control power/inch pitch
cpiroll	control power/inch Pitch
cpiyaw	Control power/inch Pitch
cppitch	Control power pitch
cproll	Control
cpyaw	Control power yaw
CQ	rotor torque coefficient
cq	coefficient of torque
CQ_sig	CQ/solidity
cqsig	Cq/sigma
CT	rotor thrust coefficient
ct	coefficient of thrust
CT_sig	CT/solidity
ctr	wing root chord
ctsig	Ct/sigma
ctsigt	ct/sigma of the tail rotor
ctw	wing tip chord
dD	differential drag, rotor blade radial segment
ddD	differential drag, rotor blade tip
ddDM	differential drag moment, rotor blade tip
ddM	differential thrust moment, rotor blade tip
ddT	differential thrust, rotor blade tip
deldv	biplane effect mutual inter. tail rotor and vert.stab
delih	span efficiency factor of the horizontal stab
deliv	span efficiency factor Of the verlical stab
deliw	span efficiency factor of the wing
delM	change in total thrust moment
delta3	tail rotor delta 3 angle
delvmax	maximum rudder deflection
desdmdq	designed pitcb damping (dM/dq)
desdndr	designed yaw damping (dN/dr)
desdrdp	design, d roll damping (dR]dp)
detafdalpfh	fuselage downwash ratio for the horizontal stab
detafdalpfw	fuselage downwash ratio) for the wing
detafdbeta	fuselage sideslip ratio (= depsilondalpha)
Dftotal	resultant of fuselage drag and aux thrust
Dfuse	total drag generated by non rotor bodies

Dhoriz	drag, horizontal tail
Dian	NOTAR boom diameter
DL	disk loading
dM	diff. thrust moment, rotor blade radial seg
DMpsi	total blade drag moment at specific azimuth
dr	rotor blade radial segment width
Drotor	differential thrust, rotor blade radial segment
dthetadM	change in cyc pitch with change in thrust moment
Dvert	drag, vertical tail
Dwing	drag, wing
e	effective hinge offset
ewing	wing efficiency factor
filename	name of input file
filename3	name of file used in plot routines
FM	figure of merit
g	acceleration due to gravity
grip	length of inner non aerodynamic portion of blade
GW	aircraft gross weight
hh	horizontal stab vertical offset
hhd	height from waterline to horizontal tail
hm	main rotor vertical offset
hmd	height from waterline to main rotor hub
Hrotor	rotor H force
hslot	total of slot heights for NOTAR boom
ht	tail rotor vertical offset
htd	height from waterline to tail rotor hub
htn	NOTAR thruster vertical offset
htnd	height from waterline to NOTAR
hv	vertical stab vertical offset
hvd	height from waterline to vertical fin
hw	wing vertical offset
hwd	height from waterline to wing
i	shaft incidence main rotor
Ib	blade flapping inertia
Ibt	tail rotor blade flapping moment of inertia
Ic	Ixx*Izz
ih	angle of incidence of horizontal stab
it	WI rotor longitudinal offset
itnd	fuselage station of NOTAR boom
iw	angle of incidence of wing
Ixx	mass moment of inertia about x axis
Ixz	mass moment of inertia about xz plane
Iyy	mass moment of inertia about y axis
Izz	mass moment of inertia about z axis
lamdaT	forward flight induced velocity parameter
lamp	lambda' inflow ratio WRT TPP
Lftotal	total lift generated by non rotor bodies
lh	horizontal stab longitudinal offset
lhd	fuselage station of horizontal tail
Lhoriz	lift, horizontal tail
lm	main rotor longitudinal offset
lmd	fuselage station of main rotor hub
lockno	main rotor lock number
locknot	tail rotor lock number
lt	tail rotor longitudinal offset
ltd	fuselage station of tail rotor hub
ltn	NOTAR thruster longitudinal offset

lttnd	fuselage station of NOTAR thruster
lv	vertical stab longitudinal offset
lvd	fuselage station of vertical tail
Lvert	lift, vertical tail
lw	wing longitudinal offset
lwd	fuselage station of wing
Lwing	lift, wing
M1c	first harmonic (cosine) thrust moment coef.
M1s	first harmonic (sine) thrust moment coefficient
Machtip	Mach number at rotor blade tip
maxr	maximum rudder deflection
mblade	mass of rotor blade
Mpsi	total blade thrust moment at specific azimuth
mu	advance ratio
mu	advance ratio
mut	tail rotor advance ratio
n	a counter in CTPLOTS.M
naz	number of azimuth sectors
nbe	number of blade elements
nt	number of tail rotor blades
ohm	omega
ohmt	tail rotor omega
omega	rotor rotational velocity
PA	pressure altitude
phi	inflow angle, rotor blade radial segment
phin	NOTAR thruster sleeve rotation angle
phitip	inflow angle, rotor blade tip
pho	roll trim attitude (euler angle)
powet	roll
Protor	power required by rotor
prpitch	Pilot rating (dM/dq)/Iyy)
prroll	Pilot rating (dR/dp)/Iy x)
pryaw	Pilot rating (dN/dr)/Izz)
psi	azimuth angle
q	dynamic pressure
q	dynamic pressure, 1/2 rho V'2
qhq	horizontal tail dynamic press ratio (qh/q)
Qrotor	rotor torque
qvin	dynamic press due to downwash at the NOTAR slots
qvq	vertical tail dynamic press ratio (qv/q)
r	radius, rotor blade radial segment
R	rotor blade radius
R	blade radius
Rbar	Reff e
RbarT	rT*Rbar
Reff	effective rotor blade radius (tip loss)
rho	ambient air density
rho	air density
rT	location of resultant thrust vector
Rt	tail rotor blade radius
Shoriz	area, horizontal tail
sidearm	maximum pe" travel or twist grip deflection
sigma	solidity
sigmat	tail rotor solidity
solidity	solidity
Svert	area, vertical tail
Swing	area, wing

swirl	mean swirl angle of main rotor meas at the NOTAR slots
T	rotor thrust
T	thrust
tailrot	value corresponding to type of tail rotor
Taux	auxiliary thrust
temp	ambient air temperature
theta	cyclic pitch
theta1c	first harmonic (cosine) of cyclic pitch
theta1s	first harmonic (sine) of cyclic pitch
thetal	main rotor blade twist
thetalt	tail rotor blade twist
thetao	collective pitch at .7 r/R
thetao	blade pitch at root
thetat	induced angle at the tip
tho	trim pitch attitude (euler angle)
Tpsi	total blade thrust at specific azimuth angle
tr	rotor blade taper ratio
Tt	thrust of the tail rotor
tv	vertical stab longitudinal offset
twist	geometric rotor blade twist
uo	x initial velocity
Up	vertical component of velocity
Uptip	vertical component of velocity at tip
Ut	horizontal component of velocity
Uttip	horizontal component of velocity at tip
V	forward velocity
v1	induced flow velocity at a hover
va1s_la	vector of a1s values for lambda variation
va1s_mu	vector of a1s values for mu variation
va1s_to	vector of a1s values for theta0 variation
vb1s_la	vector of b1s values for lambda variation
vb1s_mu	vector of b1s values for mu variation
vb1s_to	vector of b1s values for theata0 variation
vchsig_la	vector of CH/sigma values for lambda variation
vchsig_mu	vector of CH/sigma values for mu variation
vchsig_to	vector of CH/sigma values for theta0 variation
vcqsig_la	vector of CQ/sigma values for lamdda variation
vcqsig_mu	vector of CQ/sigma values for mu variation
vcqsig_to	vector of CQ/sigma values for theta0 variation
vctsig_la	vector of CT/sigma values for lambda variation
vctsig_mu	vector of CT/sigma values for mu variation
vctsig_to	vector of CT/sigma values for theta0 variation
vfvl	rotor downwash ratio for the fuselage
vhvl	rotor downwash ratio for horizontal stab
vi	induced velocity
Vinf	forward airspeed
Vlf	approximation for vi at forward airspeed
vo	y initial velocity
Vtip	tip speed
vvi_la	vector of inflow velocity for lambda variation
vvi_mu	vector of inflow velocity for mu variation
vvi_to	vector of inflow velocity for theat0 variation
vwvl	rotor downwash ratio for wing
wblade	weight of rotor blade
wo	z initial velocity
xcg	height from waterline to cg
xcouple	designed cross coupling

ycg	length from buttline to cg
yhd	length from buttline to horizontal tail
ym	main rotor lateral offset
ymd	length from buttline to rnain rotor hub
ytd	length from buttline to tail rotor hub
Ytmaxn	maximum Y force from NOTAR thruster
ytnd	length from buttline to NOTAR
Yv	side force of vertical stab
yvd	length from buttline to verticle tail
ywd	length from buttline to wing
zcg	fuselage station of cg
Zh	vertical force of horizontal stab
Zw	vertical force of win

Cruise Basic Main Rotor Derivatives

dalda	dals/dAl
daldb	dals/dBI
daldlamp	dals/d(lambda')
daldmu	dals/dmu
daldp	dals/dp
daldq	dals/dq d
daldtheto	dals/dtheta(o)
dbetdydot	d(Beta)/dYdot
dblda	dbls/dAl
dbldb	dbls/dBI
dbldlamp	dbls/d(lambda')
dbldmu	dbls/dmu
dbldp	dbls/dp
dbldq	dblsldq
db1dtheto	dblsldtheta(o)
dchsigda	d(Ch/sigma)/dals
dchsigdb	d(ch/sigma)ldb1s
dchsigdlamp	d(Ch/sigma)/d(lambda')
dchsigdmu	d(Ch/sigma)ldmu
dchsigdtheto	d(Ch/sigma)/dtheta(o)
dctsigdlamp	d(Ct/sigma)/d(lambda')
dctsigdmu	d(Ct/sigma)/drnu
dctsigdtheto	d(Ct/sigma)/dtheta(o)
dcqsigdlamp	d(Cq/sigma)/d(lambda')
dcqsigdmu	d(Cq/sigma)/dmu
dcqsigdtheto	d(Cq/sigma)/dtheta(o)
dcysigdb	d(Cylsigma)/dbls
dlampdxdot	dlambda')/dXdot
dlampdydot	d(lambda')/dYdot
dlampdzdot	d(lambda')/dZdot
dmdalsm	dM/dals main rotor stiffness
dmudxdot	dmu/dxdot
drdb1sm	dR/dB1s main rotor stiffness

Cruise Main Rotor Derivatives

dmdalm	dM/dA1
dmdblm	dM/dB1
dmdpm	dM/dp
dmdqm	dM/dq
dmdthetom	dM/dtheta(O)

dmdxdotm	dM/dXdot
dmdydotm	dM/dYdot
dmdzdotm	dM/dZdot
dndrm	dN/dr
dndtheom	dN/dtheta(O)
dndxdotm	dN/dXdot
dndzdotm	dN/dZdot
drdalm	dRIdAl
drdblm	dR/dBi
drdpm	dR/dp
drdqm	dR/dq
drdthetom	dR/dtheta(O)
drdxdotm	dR/dXdot
drdydotm	dR/dYdot
drzdotm	dR/dZdot
dxda1m	dX/dA1
dxdb1m	dXJdB1
dxdpm	dX/dp
dxdqm	dX/dq
dxdthetom	dX/dthetaO
dxdxdotm	dX/dXdot
dxdydotm	dX/dYdot
dxdzdot	dX/dZdot
dydalm	dY/dA1
dydblm	dY/dB1
dydpm	dY/dp
dydqm	dY/dq
dydtbetom	dY/dtheta(O)
dydxdotm	dY/dXdot
dydydotm	dY/dYdot
dydzdotm	dY/dZdot
dzdblm	dZ/dbls
dzdrm	dZ/dr
dzdthetom	dZ/dtheta(O)
dzdxdotm	dZ/dXdot
dzdzdotm	dZ/dZdc)t

Tail Rotor Derivatives

dctsigdlampt	d(Ct/sigma)/d(lambda’)
dctsigdmut	d(Ct/sigma)/d(mu)
dctsigdtbetot	d(Ct/sigma)/d(thetaO)
dlampdydott	d(lambda’)/dYdot
dydxdott	dY/dXdot
dydydott	dY/dYdot
dydpt	dY/dp
dydrt	dY/dr
dydthetot	dY/d(thetaO)
drdxdott	dR/dXdot
drdydott	dR/dYdot
drdpt	dR/dp
drdrt	dR/dr
drdthetot	dR/dthetaO)
dndxdott	dN/dXdot
dndydotit	dN/dYdot

Vertical Stabilizer Derivatives

Da1pvdxdotn	d(alpha vert)/dXdot
Da1pvdydotn,	d(aipba vert)/dYdot
dbetadydotv	d(beta)/dYdot
detatvdxdotv	d(eta tail rotor)/dXdot
detatvdydotv	d(eta tail rotor)/dYdot
detafdydotv	d(eta fuselage)/dYdot
drdxdotv	dR/dXdot
drdydotv	dR/dYdot
dxdxdotv	dX/dXdot
dxdydotv	dX/dYdot
dydxdotv	dY/dXdot
dydydotv	dY/dYdot
dydpv	dY/dp
dydrv	dY/dr

Horizontal Stabilizer Derivatives

dalphdxdoth	d(alpha horiz)/dXdot
dalphdzdbldoth	d(alpha horiz)/dZ double dot
dalphdzdoth	d(alpha honz)/dZdot
detamhdxdoth	d(eta main rotor)/dXdot
detafhdzdoth	d(eta fuse)/dZdot
detamhdzdoth	d(eta main rotor)/dZdot
dgamdzdoth	d(gamma)/dZdot
dxdxdoth	dX]dXdot
dxdzdbldoth	dX/dZ double dot
dxzdoth	dX/dZdot
dzdxdoth	dZ/dXdot
dzdzdbldoth	dZ/dZ double dot
dzdzdoth	dZ/dZdot

Fuselage Derivatives

the following are from the curves In Appendix A of Ref. 2:

dfdalpf	d(D/q)/d(alpha fuse)
dlqdalpf	d(L/q)/d(alpha fuse)
drnqdalpf	d(M/q)/d(alpha fuse)
dnqdbetaf	d(N/q)/d(beta)
drqdbetaf	d(RJq)/d(beta)
dsfqdbetaf	d(Y/q)/d(beta)
dalpfdxdotf	d(alpha fuse)/dXdot
dalpfdzdotf	d(alpha fuse)/dZdot
dbetadydotf	d(beta)/dYdot
detamfdxdotf	d(eta)M/dXdot
detamfdzdotf	d(eta)M/dZdot
dgamndzdotf	d(gamma)/dZdot
dmdxdotf	dM/dXdot
dnidzdotf	dM/dZdot
dndydotf	dN/dYdot
drdydotf	dR/dRdot
dxdxdotf	dX/dXdot
dxdzdotf	dX/d7 dot
dydydotf	dY/dYdot

dzdxdotf	dZ/dXdot
dxdzdot	dZ/dZdot

Hover Basic Main Rotor Derivatives

dalda	dals/dA1
daldb	dals/dB1
daldiamp	dals/d(lambda')
daldmu	dals/dmu
daldp	dals/dp
daldq	dals/dq
daldtheto	dals/dtheta(o)
dblda	dbls/dAl
dbldb	dbls/dB]
dbidlamp	dbls/d(lambda')
dbldmu	dbls/dmu
dbldp	dbls/dp
dbldq	dbls/dq
dbldtheto	dbls/dtheta(o)
dbetdydot	d(Beta)/dYdot
dchslgda	d(Ch/sigma)/dals
dchsigdb	d(Ch/sigma)/dbls
dchsigdlamp	d(Ch/sigma)/d(lambda')
dchsigdmu	d(Ch/sigma)/dmu
dchsigdtheto	d(Ch/sigma)/dtheta(o)
detsigdlamp	d(Ct/sigma)/d(lambda')
detsigdmu	d(' Ct/sigma)/dmu
dctsigdtheto	d(Ct/sigma)/dtheta(o)
dcqsigdlamp	d(Cq/sigma)/d(lambda')
dcqsigdmu	d(Cq/sigma)/dmu
dcqsigdtheto	d(Cq/sigma)/dtheta(o)
dcysigdb	d(Cy/sigma)/dbis
dlampdxdot	d(lambda')/dXdot
dlampdydot	d(lambda')/dYdot
dlampdzdot	d(lambda')/dZdot
dmdalsm	dM/dals main rotor stiffness
dmudxdot	drnu/dxdot
drdblsm	dR/dBIs main rotor stiffness

Rigging - Cockpit Stick Rigging Gains

dalmddela	dais/d(delta aileron)
dblmddele	dbls/d(delta elev)
ddelvddelp	rudder deflection per pedal travel
dphinddelp	dphi(notar thruster sleeve angle)/d(pedal)
dthetode	dtheta(0.7)/d(collective)
dthetomddele	dthetaOm/d(collective)
dthetotddelp	dtheta(0.7,tail rotor)/d(pedaJ tran,el)

Global Variables

A_MAT	A matrix for linear model
B_MAT	B matrix for linear model
AF_MAIN	Main airfoil in meshed airfoil
AF_TIP	Tip airfoil in meshed airfoil
AR	Aspect Ratio

COUNT	Counter to determine where Performance Input
FIX_TPP_VAL	Selected value for setting TPP to defined value
INTER	Iteration Interval
LAMDAT_TRIM	Trim value of lamda prime
MAXUM	Iteration End Value
MESH_VAL	Selected value when airfoil mesh option chosen
MESH_STA	r/R station where mesh occurs
MINUM	Iteration Start Value
NAME	Input .mat file name
NEW_AUX_VAL	Value of auxiliary thrust
NEW_r	Vector of user defined blade elements
NEW_TPP	Value (rads) TPP is set to for compound helo
NL_TWIST	User defined twist vector
NL_TWIST_VAL	Selected value for non linear twist
OUT_COUNT	Used to enable selection of plot routines
PICK	Iteration Method Choice (1 9)
PLOT_VALS	Values chosen for no iteration plot
RADSPC_VAL	elected value for non even blade elements
REGIME	Include HIGE Calculations Choice (1=yes, 0=no)

Structured Variables

S_H_SC_STAT	Graphics handles for Stab/Control status figure
S_MATR_VEC	Matrix/Vector structure
S_PERF_INPUT	Perf.m input structure
S_STAB_INPUT	S_STAB_INPUT_1 andS_STAB_INPUT_2
S_STAB_INPUT_1	Stab.m input structure for screen 1
S_STAB_INPUT_2	Stab.m input structure for screen 2
S_USER_INPUT	User input structure
S_FIT_TR_INPUT	Fan In Tail input structure
S_NOTAR_TR_INPUT	NOTAR input structure

Graphics Handle Variables

H_AF_MESH	Airfoil Mesh List Box
H_AL	Altitude Iteration Radio Button
H_AL_IT_P#	Altitude Iteration Plots
H_ANAL	Analysis Figure Window
H_AS	Airspeed Iteration Radio Button
H_AS_IT_P#	Airspeed Iteration Plots
H_ASPECT	Aspect Ratio Static Text Box
H_ASPECT_EDIT	Aspect Ratio Edit Text Box
H_BLD_EL	Blade Element Menu Handle
H_BK	Iteration Method << Back Push Button
H_BT	Blade Twist Iteration Radio Button
H_BT_IT_P#	Blade Twist Iteration Plots
H_BTR	Blade Taper Ratio Iteration Radio Button
H_BTR_IT_P#	Blade Taper Ratio Iteration Plots
H_check1	Save Input Data Check box
H_check2	Save Output Data Check box
H_check3	Save Matrix & Vector Data Check box
H_CID	Change Input Data Radio Button
H_CIM	Change Iteration Method Radio Button
H_CNF	Create New Radio Button
H_datain	Save Input Data Edit Box

H_dataout	Save Output Data Edit Box
H_DISK	Horiz. Tail Under Main Rotor Disk Check Box
H_EJANRAD	Exit JANRAD Radio Button
H_EREF	Edit/Run Existing File Radio Button
H_FIX_TPP	Set TPP Check Box
H_GO	Analyze Push Button
H_GW	Gross Weight Iteration Radio Button
H_GW_IT_P#	Gross Weight Iteration Plots
H_HIGE	Iteration Parameters HIGE Check box
H_inputfile	Input File Static Text box
H_IP	Iteration Parameters figure window
H_IT_BOX	Iteration Parameters Static Text Box
H_IT_METH	Iteration Method figure window
H_JAN	JANRAD 98 Figure window
H_LB	Input File List Box
H_LAT	Lateral Cyclic InputCheck Box
H_LONG	Long. Cyclic Input Check Box
H_COL	Colective Input Check Box
H_MEN	JANRAD 98 Options Menu handle
H_MESH	Mesh Parameters figure window
H_NI	No Iteration Radio Button
H_NO_IT_P#	No Iteration Plots
H_NL_TWIST	Non linear twist check box
h_opt	Performance Output JANRAD Options Menu
H_OPTIONS	Options Figure Window
H_outputfile	Output File Static Text box
H_P	Performace Radio Button
H_PED	Pedal Input Check Box
H_POP	Airfoil List Box
H_PERF_IN	Performance Input Figure Window
H_PERF_OUT	Performance Output Figure Window
H_PRDA	Perform Rotor Dynamics Radio Button
H_printin	Print Input File Check Box
H_printout	Print Output File Check Box
H_printvec	Print Matrix & Vector File Check Box
H_PSCA	Perform Stability and Control Radio Button
H_RBR_IT_P#	Rotor Radius Iteration plots
H_RBS_IT_P#	Rotor Speed Iteration plots
H_RD	Rotor Dynamics Radio Button
H_RES	Resume Push Button
H_RTB	Return to Beginning Radio Button
H_RUPT	Interrupt Push Button
H_SAC	Stability and Control Radio Button
H_SOT	Start of Taper Iteration Radio Button
H_SOT_IT_P#	Start of Taper Iteration Plots
H_SC_SAVE	Stab/Conrrol Save File Output Figure Window
H_SAVE_CON	Stab/Control Save File Output Confirm Window
H_STAB_IN1	Stab/Control Input Figure page 1
H_STAB_IN2	Stab/Control Input Figure page 2
H_STAB_OUT	Stab/Control output figure window
H_STATUS	Top Analysis Status Static Text Box
H_STATUS1	Middle Analysis Status Static Text Box
H_STATUS2	Middle Analysis Status Static Text Box
H_STATUS3	Bottom Analysis Status Static Text Box
H_STBY_SCRN	Stab/Conrol Standby Screen Figure Window
H_vecdata	Save Matrix & Vector Data Edit Box
H_vecfile	Vector File Static Text box

H_WORK	Working Directory Edit Box
H_WSA	Wing Span Area Iteration Radio Button
H_WSA_IT_P#	Wing Span Area Iteration Plots

APPENDIX D. MATLAB M-FILES

1. about_janrad.m

```
function about_janrad()
% This is the machine-generated representation of a MATLAB object
% and its children.  Note that handle values may change when these
% objects are re-created. This may cause problems with some callbacks.
% The command syntax may be supported in the future, but is currently
% incomplete and subject to change.
%
% To re-open this system, just type the name of the m-file at the MATLAB
% prompt. The M-file and its associtated MAT-file must be on your path.

% This function generates a screen listing all contributers to the
JANRAD
% program.

% Modified for JANRAD 6.0 by LT D.A. Heathorn

load about_janrad

a = figure('Units','normalized', ...
      'Color',[0.8 0.8 0.8], ...
      'Colormap',mat0, ...
      'MenuBar','none', ...
      'Name','About JANRAD 98', ...
      'NumberTitle','off', ...
      'PointerShapeCData',mat1, ...
      'Position',[0.1075 0.13 0.6675 0.77], ...
   'Tag','Fig1');
b = uicontrol('Parent',a, ...
      'Units','normalized', ...
      'BackgroundColor',[0.752941 0.752941 0.752941], ...
      'Position',[0.00749064 0.0194805 0.975655 0.963203], ...
      'Style','frame', ...
      'Tag','Frame1');
b = uicontrol('Parent',a, ...
      'Units','normalized', ...
      'BackgroundColor',[0.752941 0.752941 0.752941], ...
      'Position',[0.0299625 0.348485 0.930712 0.316017], ...
      'Style','frame', ...
      'Tag','Frame2');

b = uicontrol('Parent',a, ...
      'Units','normalized', ...
      'BackgroundColor',[0.752941 0.752941 0.752941], ...
      'FontSize',14, ...
      'FontWeight','bold', ...
      'Position',[0.0189125 0.85119 0.959811 0.122024], ...
      'String','Joint Army/Navy Rotorcraft Analysis and Design   JANRAD
98', ...
      'Style','text', ...
      'Tag','StaticText1');
```

```
b = uicontrol('Parent',a, ...
      'Units','normalized', ...
      'BackgroundColor',[0.752941 0.752941 0.752941], ...
      'FontSize',10, ...
      'FontWeight','bold', ...
      'Position',[0.230428 0.743017 0.536189 0.104283], ...
      'String','Naval Postgraduate School  Monterey,  California', ...
      'Style','text', ...
      'Tag','StaticText2');
b = uicontrol('Parent',a, ...
      'Units','normalized', ...
      'BackgroundColor',[0.752941 0.752941 0.752941], ...
      'FontSize',10, ...
      'FontWeight','demi', ...
      'Position',[0.401773 0.674115 0.192024 0.0595903], ...
      'String','March 1999', ...
      'Style','text', ...
      'Tag','StaticText3');
b = uicontrol('Parent',a, ...
      'Units','normalized', ...
      'BackgroundColor',[0.752941 0.752941 0.752941], ...
      'Position',[0.0355805 0.582251 0.911985 0.0367965], ...
      'String','Version 5.0 Designer: LCDR. William L. Hucke,  USCG',
...
      'Style','text', ...
      'Tag','StaticText4');
b = uicontrol('Parent',a, ...
      'Units','normalized', ...
      'BackgroundColor',[0.752941 0.752941 0.752941], ...
      'Position',[0.0318352 0.536797 0.913858 0.0411255], ...
      'String','Version 4.0 Designer: LCDR. Chris F. Lapacik,  USN', ...
      'Style','text', ...
      'Tag','StaticText4');
b = uicontrol('Parent',a, ...
      'Units','normalized', ...
      'BackgroundColor',[0.752941 0.752941 0.752941], ...
      'Position',[0.0318352 0.443723 0.913858 0.04329], ...
      'String','Version 3.0 Designer: LT. Dave Eccles,  USN', ...
      'Style','text', ...
      'Tag','StaticText5');
b = uicontrol('Parent',a, ...
      'Units','normalized', ...
      'BackgroundColor',[0.752941 0.752941 0.752941], ...
      'Position',[0.0318352 0.400433 0.913858 0.0411255], ...
      'String','Version 2.0 Designer: LT. Dale Feddersen,  USN', ...
      'Style','text', ...
      'Tag','StaticText1');
b = uicontrol('Parent',a, ...
      'Units','normalized', ...
      'BackgroundColor',[0.752941 0.752941 0.752941], ...
      'Position',[0.0355805 0.491342 0.913858 0.0411255], ...
      'String','Version 3.1 Designer: LT. Dan Hiatt,  USN', ...
      'Style','text', ...
      'Tag','StaticText6');
b = uicontrol('Parent',a, ...
      'Units','normalized', ...
      'BackgroundColor',[0.752941 0.752941 0.752941], ...
```

```
      'Position',[0.0337079 0.354978 0.913858 0.0411255], ...
      'String','Version 1.0 Designers:  MAJ Bob Nicholson,  USA & MAJ.
Walter Wirth,  USA', ...
      'Style','text', ...
      'Tag','StaticText7');
b = uicontrol('Parent',a, ...
      'Units','normalized', ...
      'BackgroundColor',[0.752941 0.752941 0.752941], ...
      'HorizontalAlignment','left', ...
      'Position',[0.0945347 0.148976 0.807976 0.189944], ...
      'String',mat2, ...
      'Style','text', ...
      'Tag','StaticText8');
b = uicontrol('Parent',a, ...
      'Units','normalized', ...
      'Callback','close (gcf)', ...
      'FontSize',14, ...
      'FontWeight','bold', ...
      'Position',[0.408983 0.0505952 0.177305 0.0892857], ...
      'String','OK', ...
      'Tag','Pushbutton1');
b = uicontrol('Parent',a, ...
      'Units','normalized', ...
      'BackgroundColor',[0.752941 0.752941 0.752941], ...
      'Position',[0.0318352 0.623377 0.911985 0.0367965], ...
      'String','Version 6.0 Desienger:  LT David A. Heathorn, USN', ...
      'Style','text', ...
      'Tag','StaticText4');
```

2. Bode_plotter

```
% Bode_plotter.m

% M-file called by time_freq_resp_fcn.m to plot Bode Diagrams for
specified input and output.

% Created for JANRAD version 6.0 by LT David A. Heathorn

global Amat Bmat u C


D=0;

if u(1)==1

   for j=1:length(C)
      if C(j,j)==1
         figure
         bode(Amat, Bmat(:,1), C(j,:), D)
        if j==1
            title('Response of x-velocity (u) to  Longitudinal Cyclic
Input')

         elseif j==2
            title('Response of z-velocity (w) to Longitudinal Cyclic
Input')

         elseif j==3
            title('Response of Pitch Rate (q) to Longitudinal Cyclic
Input')

         elseif j==4
            title('Response of Pitch Angle (theta) to Longitudinal
Cyclic Input')

         elseif j==5
            title('Response of y-veloctiy (v) to Longitudinal Cyclic
Input')

         elseif j==6
            title('Response of Roll Rate (p) to Longitudinal Cyclic
Input')

         elseif j==7
            title('Response of Roll Angle (phi) to Longitudinal Cyclic
Input')

         elseif j==8
            title('Response of Yaw Rate (r) to Longitudinal Cyclic
Input')

         elseif j==9
            title('Response of Yaw Angle (psi) to Longitudinal Cyclic
Input')

         end
```

```
      end
   end

elseif u(2)==1

   for j=1:length(C)
      if C(j,j)==1
         figure
         bode(Amat, Bmat(:,2), C(j,:), D)

         if j==1
            title('Response of x-velocity (u) to Collective  Input')

         elseif j==2
            title('Response of z-velocity (w) to Collective Input')

         elseif j==3
            title('Response of Pitch Rate (q) to Collective Input')

         elseif j==4
            title('Response of Pitch Angle (theta) to Collective Input')

         elseif j==5
            title('Response of y-veloctiy (v) to Collective Input')

         elseif j==6
            title('Response of Roll Rate (p) to Collective Input')

         elseif j==7
            title('Response of Roll Angle (phi) to Collective Input')

         elseif j==8
            title('Response of Yaw Rate (r) to Collective Input')

         elseif j==9
            title('Response of Yaw Angle (psi) to Collective Input')

         end
      end
   end


elseif u(3)==1
   for j=1:length(C)
      if C(j,j)==1
         figure
         bode(Amat, Bmat(:,3), C(j,:), D)
         if j==1
            title('Response of x-velocity (u) to Lateral Cyclic Input')

         elseif j==2
            title('Response of z-velocity (w) to Lateral Cyclic Input')

         elseif j==3
            title('Response of Pitch Rate (q) to Lateral Cyclic Input')

         elseif j==4
```

```
            title('Response of Pitch Angle (theta) to Lateral Cyclic
Input')

         elseif j==5
            title('Response of y-veloctiy (v) to Lateral Cyclic Input')

         elseif j==6
            title('Response of Roll Rate (p) to Lateral Cyclic Input')

         elseif j==7
            title('Response of Roll Angle (phi) to Lateral Cyclic
Input')

         elseif j==8
            title('Response of Yaw Rate (r) to Lateral Cyclic Input')

         elseif j==9
            title('Response of Yaw Angle (psi) to Lateral Cyclic Input')

         end
      end
   end


elseif u(4)==1

   for j=1:length(C)
      if C(j,j)==1
         figure
         bode(Amat, Bmat(:,4), C(j,:), D)
         if j==1
            title('Response of x-velocity (u) to Pedal Input')

         elseif j==2
            title('Response of z-velocity (w) to Pedal Input')

         elseif j==3
            title('Response of Pitch Rate (q) to Pedal Input')

         elseif j==4
            title('Response of Pitch Angle (theta) to Pedal Input')

         elseif j==5
            title('Response of y-veloctiy (v) to Pedal Input')

         elseif j==6
            title('Response of Roll Rate (p) to Pedal Input')

         elseif j==7
            title('Response of Roll Angle (phi) to Pedal Input')

         elseif j==8
            title('Response of Yaw Rate (r) to Pedal Input')

         elseif j==9
            title('Response of Yaw Angle (psi) to Pedal Input')
```

```
            end
        end
    end

end
```

3. Cbodygroup

```
% Cbodygrp.m
% CALLED BY Cruise.m
% Computes the stability derivatives of the fuselage, wing, verticle
% fin and horizontal stabalizer in cruise flight.
%
%
% Compute the stability derivatives of the fuselage
%
% Modified for JANRAD version 6.0 by LT David A. Heathorn
%
dgamdzdotf=-1/V;
dbetadydotf=1/V;
detamfdxdotf=vfv1/(4*q*pi*R^2)*(-dzdxdotm-2*T/V);
detamfdzdotf=-vfv1/(4*q*pi*R^2)*dzdzdotm;
dalpfdxdotf=-detamfdxdotf;
dalpfdzdotf=-(detamfdzdotf+dgamdzdotf);
%
%  the folowing are from the curves in appendix A of Prouty:
%  and can be modified if necessary
dfdalpf=-2;
dlqdalpf=74.5;
dsfqdbetaf=-220;
dmqdalpf=1780;
dmqdalpf_intercept=-175;
dnqdbetaf=-820;
drqdbetaf=230;
%
dxdxdotf=-2/V*Afh*q;
dxdzdotf=(-Afh*q-q*dfdalpf)*dalpfdzdotf;
dydydotf=1/V*(q*dsfqdbetaf-Afh*q);
dzdxdotf=(2/V)*Afh*q;
dzdzdotf=(-Afh*q-q*dlqdalpf)*dalpfdzdotf;
drdydotf=q*drqdbetaf*dbetadydotf;

Mf_bar=((dmqdalpf*-(im+alphaT))+dmqdalpf_intercept)*q;
dmdxdotf=q*dmqdalpf*dalpfdxdotf+(2*Mf_bar/V);

dmdzdotf=q*dmqdalpf*dalpfdzdotf;
dndydotf=q*dnqdbetaf*dbetadydotf;
%
% Compute the stability derivatives of the wing
%
if wing==1
  dgamdzdotw=-1/V;
  detamhdxdotw=vwv1/(4*q*pi*R^2)*(-dzdxdotm-2*T/V);
  detamhdzdotw=-vwv1/(4*q*pi*R^2)*dzdzdotm;
  detafhdzdotw=detafdalpfw*(1/(4*q*pi*R^2)*dzdzdotm-dgamdzdotw);
  dalphdxdotw=-detamhdxdotw;
  dalphdzdotw=-(detamhdzdotw+detafhdzdotw+dgamdzdotw);
  dalphdzdbldotw=-detamhdzdotw*lw/V;
  %
  dxdxdotw=-2*cdow;
  dxdzdotw=q*Aw*aw*((alpw-alplow)*(1-2*aw*(1+deliw)*Aw/pi/bw^2)+alpw-
iw)*...
    dalphdzdotw;
```

```
  dxdzdbldotw=dxdzdotw*dalphdzdbldotw/dalphdzdotw;
  dzdxdotw=2/V*Zw-q*Aw*aw*(1+aw*(1+deliw)*Aw/pi/bw^2*(2*(alpw-
alplow)*...
    (alpw-iw)+(alpw-alplow)^2)+cdow)*dalphdxdotw;
  dzdzdotw=-q*Aw*aw(1+aw*(1+deliw)*Aw/pi/bw^2*(2*(alpw-alplow)*...
    (alpw-iw)+(alpw*alplow)^2)+cdow)*dalphdzdotw;
  dzdzdbldotw=dzdzdotw*dalphdzdbldotw/dalphdzdotw;
  %
  dzdqw=dzdzdotw*lw;
  dzdqw=dzdzdotw*lw;
  dmdxdotw=-dxdxdotw*hw+dzdxdotw*lw;
  dmdzdotw=-dxdzdotw*hw+dzdzdotw*lw;
  dmdzdbldotw=-dxdzdbldotw*hw+dzdzdbldotw*lw;
  dmdqw=dzdqw*lh;
  %
  drdrw=q*Aw*bw*bw*aw*alpw/4/V/V/Ixx;
  drdpw=-q*Aw*bw*bw/4/V/V/Ixx*pi*(1+3*ctw/crw)/(6*(1+ctw/crw));
else
  %
  % Zero stability derivatives of the wing when no wing is installed
  %
  dxdxdotw=0;dxdzdotw=0;dxdzdbldotw=0;dzdxdotw=0;
  dzdzdotw=0;dzdzdbldotw=0;
  dzdqw=0;dzdqw=0;dmdxdotw=0;dmdzdotw=0;
  dmdzdbldotw=0;dmdqw=0;
  drdrw=0;drdpw=0;
end
%
% Compute the stability derivatives of the verticle fin
%
detafdbeta=.06; % assumed to be .06 because of little study of effect
dbetadydotv=1/V;
detatvdxdotv=-1/(4*qvq*q*Av)*(dydxdott-2*Tt/V);
dalpvdxdotv=detatvdxdotv;
detatvdydotv=dydydott/(4*qvq*q*At);
detafdydotv=detafdbeta*dbetadydotv;
dalpvdydotv=-(dbetadydotv+detatvdydotv+detafdydotv);
%
dxdxdotv=0;
dxdydotv=0;
dydxdotv=2/V*Yv+qvq*q*Av*av*dalpvdxdotv;
dydydotv=1/(1-deldv/Yv)*...
  (qvq*q*Av*av*dalpvdydotv+deldv*(-2/V+1/Tt*dydydott));
%
dydpv=dydydotv*hv;
dydrv=-dydydotv*lv;
drdxdotv=dydxdotv*hv;
drdydotv=dydydotv*hv;
drdpv=dydpv*hv;
drdrv=dydrv*hv;
dndxdotv=-dydxdotv*lv;
dndydotv=-dydydotv*lv;
dndpv=-dydpv*lv;
dndrv=-dydrv*lv;
%
dyddelv=q*qvq*Av*clvertmax/delvmax;
drddelv=dyddelv*hv;
```

```
dnddelv=-dyddelv*lv;
%
% Compute the stability derivatives of the horizontal stabalizer
%
dgamdzdoth=-1/V;
detamhdxdoth=vhv1/(4*q*pi*R^2)*(-dzdxdotm-2*T*cos(altpp)/V);
detamhdzdoth=-vhv1/(4*q*pi*R^2)*dzdzdotm;
detafhdzdoth=detafdalpfh*(1/(4*q*pi*R^2)*dzdzdotm-dgamdzdoth);
dalphdxdoth=-detamhdxdoth;
dalphdzdoth=-(detamhdzdoth+detafhdzdoth+dgamdzdoth);
dalphdzdbldoth=-detamhdzdoth*lh/V;
%
etamh=vhv1*v1/V;
etafh=detafdalpfh*tho;
Xh=(Lhoriz*sin(tho-(etamh+etafh+gamc))-Dhoriz*cos(tho-
(etafh+etafh+gamc)));
dxdxdoth=2/V*Xh+qhq*q*Ah*ah*((alph-alploh)*(1-
2*ah*(1+delih)*Ah/pi/bh^2)+alph-ih)*...
  dalphdxdoth;
dxdzdoth=qhq*q*Ah*ah*((alph-alploh)*(1-2*ah*(1+delih)*Ah/pi/bh^2)+alph-
ih)*...
  dalphdzdoth;
dxdzdbldoth=dxdzdoth*dalphdzdbldoth/dalphdzdoth;

Zh=(-Lhoriz*cos(tho-(etamh+etafh+gamc))-Dhoriz*cos(tho-
(etafh+etafh+gamc)));

dzdxdoth=(2/V*(Zh))-qhq*q*Ah*ah*(1+ah*(1+delih)*Ah/pi/bh^2*(2*(alph-
alploh)*...
   (alph-ih)+(alph-alploh)^2)+cdoh)*dalphdxdoth;
dzdzdoth=-abs(-qhq*q*Ah*ah*(1+ah*(1+delih)*Ah/pi/bh^2*(2*(alph-
alploh)*...
  (alph-ih)+(alph*alploh)^2)+cdoh)*dalphdzdoth);
dzdzdbldoth=dzdzdoth*dalphdzdbldoth/dalphdzdoth;
%
dzdqh=dzdzdoth*lh;
dzdqh=dzdzdoth*lh;
dmdxdoth=-dxdxdoth*hh+dzdxdoth*lh;
dmdzdoth=-dxdzdoth*hh+dzdzdoth*lh;
dmdzdbldoth=-dxdzdbldoth*hh+dzdzdbldoth*lh;
dmdqh=dzdqh*lh;
%
% return to CRUISE.M
```

4. Cmrgrp.m

```
% Cmgrp.m
% CALLED BY Cruise.m
% Computes the basic MR derivatives in cruise flight
% Computes the stability derivatives of the main rotor in cruise flight.
% Uses data loaded in the workspace by JANRAD.M and STAB.M
%
% Compute the basic mainrotor derivatives.
%
% Modified for JANRAD version 6.0 by LT David A. Heathorn
%
dmudxdot=1/ohm/R ;
dlampdxdot=(1/ohm/R)*(-altpp-(sigma/2/mu)*(dctsigdmu-ctsig/mu)) ;
dlampdzdot=1/(ohm*R*(1+dctsigdlamp*sigma/2/mu)) ;
dbetdydot=1/V ;
dchsigda=ctsig+a/8*lamp ;
dcysigdb=dchsigda ;
da1dq=-16/(lockno*ohm*(1-e/R)^2*(1-mu*mu/2))-...
12*e/R/(lockno*ohm*(1-e/R)^3*(1-mu^4/4)) ;
da1dp=1/(ohm*(1-mu^2/2))-192*e/R/(lockno^2*ohm*(1-e/R)^5*(1-mu^4/4)) ;
da1da=12*e/R*(1+mu^2/2)/(lockno*(1-e/R)^3*(1-mu^4/4)) ;
da1db=-1*(1+3/2*mu^2)/(1-mu^2/2) ;
db1dq=-inv(ohm*(1+mu^2/2))+192*e/R/(lockno^2*ohm*(1-e/R)^5*(1-mu^4/4)) ;
db1dp=-16/(lockno*ohm*(1-e/R)^2*(1+mu*mu/2))-...
12*e/R/(lockno*ohm*(1-e/R)^3*(1-mu^4/4)) ;
db1da=1 ;
db1db=12*e/R*(1+3/2*mu^2)/(lockno*(1-e/R)^3*(1-mu^4/4)) ;
dmda1sm=3/4*e/R*Ab*rho*R*(ohm*R)^2*a/lockno ;
drdb1sm=dmda1sm ;
%
% Compute the mainrotor stability derivatives.

%
dxdxdotm=-
rho*Ab*(ohm*R)^2*((dchsigdmu+dchsigda*da1dmu+(a1s+im)*dctsigdmu)...
*dmudxdot+((dchsigdlamp+dchsigda*da1dlamp+(a1s+im)*dctsigdlamp)*dlampdxd
ot)) ;
dxdydotm=rho*Ab*(ohm*R)^2*ctsig*(A1-b1s)*dbetdydot ;
dxdzdotm=-
rho*Ab*(ohm*R)^2*(dchsigdlamp+(ctsig*da1dlamp)+(a1s+im)*dctsigdlamp)...
*dlampdzdot ;
dxdqm=-rho*Ab*(ohm*R)^2*dchsigda*da1dq-dxdxdotm*hm ;
dxdpm=-rho*Ab*(ohm*R)^2*dchsigda*da1dp-dxdydotm*hm ;
dxdthetom=-rho*Ab*(ohm*R)^2*(dchsigdtheto+dchsigda*da1dtheto+...
(a1s+im)*dctsigdtheto) ;
dxda1m=-rho*Ab*(ohm*R)^2*dchsigda*da1da ;
dxdb1m=-rho*Ab*(ohm*R)^2*dchsigda*da1db ;
dydxdotm=rho*Ab*(ohm*R)^2*(dcysigdb*db1dmu+b1s*dctsigdmu)*dmudxdot ;
dydydotm=-rho*Ab*(ohm*R)^2*(chsig+ctsig*(B1+a1s))*dbetdydot ;
dydzdotm=rho*Ab*(ohm*R)^2*db1dlamp*dcysigdb*dlampdzdot ;
dydqm=rho*Ab*(ohm*R)^2*dcysigdb*db1dq+dydxdotm*hm ;
dydpm=rho*Ab*(ohm*R)^2*dcysigdb*db1dp+dydydotm*hm ;
dydthetom=rho*Ab*(ohm*R)^2*(dcysigdb*db1dtheto+b1s*dctsigdtheto) ;
dyda1m=rho*Ab*(ohm*R)^2*dcysigdb*db1da ;
dydb1m=rho*Ab*(ohm*R)^2*dcysigdb*db1db ;
dzdxdotm=-rho*Ab*(ohm*R)^2*(dctsigdmu*dmudxdot+dctsigdlamp*dlampdxdot) ;
```

```
dzdzdotm=-rho*Ab*(ohm*R)^2*dctsigdlamp*dlampdzdot ;
dzdrm=2/ohm*rho*Ab*(ohm*R)^2*ctsig ;
dzdthetom=-rho*Ab*(ohm*R)^2*dctsigdtheto ;
dzdb1m=-rho*Ab*(ohm*R)^2*dctsigdlamp*inv(da1dlamp)*da1db ;
drdxdotm=drdb1sm*db1dmu*dmudxdot+dydxdotm*hm+dzdxdotm*ym ;
drdydotm=-drdb1sm*(B1+a1s)*dbetdydot+dydydotm*hm ;
drdzdotm=drdb1sm*db1dlamp*dlampdzdot+dydzdotm*hm+dzdzdotm*ym ;
drdqm=drdb1sm*db1dq+dydqm*hm ;
drdpm=drdb1sm*db1dp+dydpm*hm ;
drdthetom=drdb1sm*db1dtheto+dydthetom*hm+dzdthetom*ym ;
drda1m=drdb1sm*db1da+dyda1m*hm ;
drdb1m=drdb1sm*db1db+dydb1m*hm ;
dmdxdotm=dmda1sm*(da1dmu*dmudxdot+da1dlamp*dlampdxdot)+dxdxdotm*hm+dzdxd
otm*lm ;
dmdydotm=-dxdydotm*hm+dmda1sm*(A1-b1s)*dbetdydot ;
dmdzdotm=dmda1sm*da1dlamp*dlampdzdot-dxdzdotm*hm+dzdzdotm*lm ;
dmdqm=dmda1sm*da1dq-dxdqm*hm ;
dmdpm=dmda1sm*da1dp+dxdpm*hm ;
dmdthetom=dmda1sm*da1dtheto-dxdthetom*hm+dzdthetom*lm ;
dmda1m=dmda1sm*da1da-dxda1m*hm ;
dmdb1m=dmda1sm*da1db-dxdb1m*hm ;
dndxdotm=rho*Ab*(ohm*R)^2*R*(dcqsigdmu*dmudxdot+dcqsigdlamp*dlampdxdot)
;

dndzdotm=rho*Ab*(ohm*R)^2*R*dcqsigdlamp*dlampdzdot ;
dndrm=-2/ohm*rho*Ab*(ohm*R)^2*R*cqsig ;
dndthetom=rho*Ab*(ohm*R)^2*R*dcqsigdtheto ;
%
% return to CRUISE.M
```

5. Cruise.m

```
% Cruise.m
%
% Computes the stability derivatives in cruise flight.
% calls the following subroutines
%
% Cmrgrp
% Ctrgrp
% Cfusegrp
% Trim
% Dctplots or Dctmats
%
% evaluate basic stability derivatives
%
% MAIN ROTOR
% dctsigdmu   dctsigdtheto   dctsigdlamp
% dchsigdmu   dchsigdtheto   dchsigdlamp
% dcqsigdmu   dcqsigdtheto   dcqsigdlamp
% da1dmu      da1dtheto      da1dlamp
% db1dmu      db1dtheto      db1dlamp
%
% TAIL ROTOR
% dctsigdlampt dctsigdmut dctsigdthetot
%

% Modified for JANRAD version 6.0  by LT David A. Heathorn

global S_SC_INPUT_1 S_SC_INPUT_2 S_PERF_INPUT S_PERF_OUTPUT H_S_SC_STAT
S_H_SC_STAT Amat Bmat



% Unstructure input variables

S_STAB_INPUT_1=S_SC_INPUT_1;
S_STAB_INPUT_2=S_SC_INPUT_2;

unstructure
unstructure_stab_input_1
unstructure_stab_input_2
unstructure2

H_SC_STAT=S_H_SC_STAT.h_sc_stat;
H_STATUS=S_H_SC_STAT.h_status;
H_STATUS1=S_H_SC_STAT.h_status1;
H_STATUS2=S_H_SC_STAT.h_status2;
H_STATUS3=S_H_SC_STAT.h_status3;
H_STATUS4=S_H_SC_STAT.h_status4;
H_BK=S_H_SC_STAT.h_bk;
H_GO=S_H_SC_STAT.h_go;
H_RUPT=S_H_SC_STAT.h_rupt;
H_RES=S_H_SC_STAT.h_res;
H_MEN=S_H_SC_STAT.h_men;


% VARIABLE UNIT CONVERSION
```

```
k=1;
im=im/57.3;
delta3=delta3/57.3;
theta1t=theta1t/57.3;
alplov=alplov/57.3;
alploh=alploh/57.3;
ih=ih/57.3;
alplow=alplow/57.3;
iw=iw/57.3;
swirl=swirl/57.3;
db1mddele=db1mddele/57.3;
da1mddela=da1mddela/57.3;
dthetomddelc=dthetomddelc/57.3;
dthetotddelp=dthetotddelp/57.3;
sidearm=sidearm*2;
theta1c=theta1c/57.3;
theta1s=theta1s/57.3;
thetao=thetao/57.3;
alphaT=alphaT/57.3;
betao=betao/57.3;



if sidearm==0
   sidearm=1e3;
end
dphinddelp=pi/sidearm;
sidearm=pi/(sidearm*2);

theta1=-twist;

set(H_STATUS2,'STRING', 'CRUISE ROUTINE')
set(H_STATUS4,'STRING', 'FIRST OF THREE PARAMATER VARIATIONS')
tic
set(H_STATUS1,'STRING', 'START ELAPSED TIME')
pause(3)

% vary mu

set(H_STATUS4,'STRING', 'VARYING MU')
stab_calc_mu

% vary thetao


set(H_STATUS4,'STRING', 'VARYING THETAO')
stab_calc_to
% vary lambda prime

set(H_STATUS4,'STRING', 'VARYING LAMDA PRIME')
stab_calc_la

% Calculating derivative from vectors.

set(H_STATUS4,'STRING', 'COMPUTING DERIVATIVES')
```

```
dctplots

if tailrot==1
   dctplott
end

% CONFIGURATION CALCULATION

set(H_STATUS4,'STRING', 'CALCULATING MATICIES')

unstructure2
theta1c=theta1c/57.3;
theta1s=theta1s/57.3;
thetao=thetao/57.3;
thetao=thetao-.7*theta1;
ctsig=vctsig_mu(1,2);
chsig=vchsig_mu(1,2);
cqsig=vcqsig_mu(1,2);
alphaT=alphaT/57.3;
altpp=alphaT;
A1=-theta1c;
B1=-theta1s;
ao=betao;
v1=vvi_mu(1,2);
V=Vinf;
sigma=solidity;

% Main Rotor
ohm=omega;
lv=lvd-xcg;
g=32.2;
A=pi*R*R;

hm=hmd-zcg;
ym=ymd-ycg;
lm=lmd-xcg;
c=(sum(cblade)/length(r));
Ab=c*R*b;
lamp=vlamp_la(1,2);
mu=vmu_mu(1,2);
ct=CT;
f=Afv;
thetat=4/a*ctsig+sqrt(solidity*ctsig/2);
lockno=rho*a*c*R^4/Ib;
theta75=thetao+.75*theta1;
a1s=0;
b1s=0;
m=GW/g;
Ic=Ixx*Izz;
lh=lhd-xcg;
hh=hhd-zcg;
yh=yhd-ycg;
hv=hvd-zcg;
yv=yvd-ycg;
delvmax=pi/4;    %est max verticle fin defl before stall
if maxr==0
```

```
   ddelvddelp=0;
else
   ddelvddelp=delvmax/maxr;   % verticle fin deflection/maxr
end
if tailrot==3;
   % NOTAR
   v1max=1.2*v1;     %  downwash at the boom slots (NOTAR)
   ltn=lttnd-xcg;
   ln=ltnd-xcg;
   yn=ytnd-xcg;
   hn=htnd-zcg;
   htn=hn;
   Tt=(cqsig*solidity*rho*A*(ohm*R)^2*R-Lvert*lv)/ltn;
   cmun=.55;  % a typical value
   bn=.5*R;
   hslot=.028*dian/2;
   arn=bn/dian;
   At=pi*dian*dian/4;
   Rt=dian/2;
end
if Swing<.1
   wing=0;
else
   Aw=Swing;
   lw=lwd-xcg;
   hw=hwd-zcg;
   yw=ywd-ycg;
   alpw=CLwing/aw;
   deliw=ewing;
   Zw=Lwing;
   bw=bwing;
   cdow=CDowing;
   wing=1;
end
Av=Svert;
bv=bvert;
Ah=Shoriz;
bh=bhoriz;
cdov=CDovert;
cdoh=CDohoriz;
deliv=ewing;
delih=ewing;
alph=CLhoriz/ah;
Zh=Lhoriz;
Yv=Lvert;
if Yv<.1;Yv=.1;end;
deldv=1.72*(1-bv/(20+Rt))*(2*Yv/(2*Rt+bv));


% trim conditions
tho=0;
pho=0;
vo=0;
uo=Vinf*cos(tho);
wo=uo*tan(tho);
gamc=0;
```

```
% STABILITY CALCULATION

Cmrgrp
Ctrgrp
Cbodygrp

% computation of A,B,C,D matrices

A11= (dxdxdotm+dxdxdoth+dxdxdotv+dxdxdotf+dxdxdotw)/m;
A12= (dxdzdotm+dxdzdoth+dxdzdotf+dxdzdotw)/m;
A13= (dxdqm)/m-wo;
A14= -g*cos(tho);
A15= (dxdydotm+dxdydotv)/m;
A16= dxdpm/m;
A17= 0;
A18= 0;
%
A21= (dzdxdotm+dzdxdoth+dzdxdotf+dzdxdotw)/m;
A22= (dzdzdotm+dzdzdoth+dzdzdotf+dzdzdotw)/m;
A23= (dzdqh+dzdqw)/m+uo;
A24= -g*cos(pho)*sin(tho);
A25= 0;
A26= 0;
A27= -g*sin(pho)*cos(tho);
A28= dzdrm/m;
%
A31= (dmdxdotm+dmdxdoth+dmdxdotf+dmdxdotw)/Iyy;
A32= (dmdzdotm+dmdzdoth+dmdzdotf+dmdzdotw)/Iyy;
A33= (dmdqm+dmdqh+dmdqw)/Iyy;
A34= 0;
A35= dmdydotm/Iyy;
A36= (dmdpm)/Iyy;
A37= 0;
A38= 0;
%
A41= 0;
A42= 0;
A43= cos(pho);
A44= 0;
A45= 0;
A46= 0;
A47= 0;
A48= -sin(pho);
%
A51= (dydxdotm+dydxdott+dydxdotn+dydxdotv)/m;
A52= (dydzdotm+dydzdotn)/m;
A53= dydqm/m;
A54= -g*sin(pho)*sin(tho);
A55= (dydydotm+dydydott+dydydotn+dydydotv+dydydotf)/m;
A56= (dydpm+dydpt+dydpn+dydpv)/m+wo;
A57= g*cos(pho)*cos(tho);
A58= (dydrt+dydrn+dydrv)/m-uo;
%
A61= (Izz*(drdxdotm+drdxdott+drdxdotn+drdxdotv)...
   +Ixz*(dndxdotm+dndxdott+dndxdotn+dndxdotv))/Ic;
A62= (Izz*(drdzdotm+drdzdotn)+Ixz*(dndzdotm+dndzdotn))/Ic;
A63= (Izz*(drdqm)+Ixz*(0))/Ic;
```

```
A64= 0;
A65= (Izz*(drdydotm+drdydott+drdydotn+drdydotv+drdydotf)...
   +Ixz*(dndydott+dndydotn+dndydotv+dndydotf))/Ic;
A66= (Izz*(drdpm+drdpt+drdpn+drdpv+drdpw)+Ixz*(dndpt+dndpn+dndpv))/Ic;
A67= 0;
A68= (Izz*(drdrt+drdrn+drdrv+drdrw)+Ixz*(dndrm+dndrt+dndrn+dndrv))/Ic;
%
A71= 0;
A72= 0;
A73= sin(pho)*tan(tho);
A74= 0;
A75= 0;
A76= 1;
A77= 0;
A78= cos(pho)*tan(tho);
%
A81= (Ixz*(drdxdotm+drdxdott+drdxdotn+drdxdotv)...
   +Ixx*(dndxdotm+dndxdott+dndxdotn+dndxdotv))/Ic;
A82= (Ixz*(drdzdotm+drdzdotn)+Ixx*(dndzdotm+dndzdotn))/Ic;
A83= (Ixz*(drdqm)+Ixx*(0))/Ic;
A84= 0;
A85= (Ixz*(drdydotm+drdydott+drdydotn+drdydotv+drdydotf)...
   +Ixx*(dndydott+dndydotn+dndydotv+dndydotf))/Ic;
A86= (Ixz*(drdpm+drdpt+drdpn+drdpv+drdpw)+Ixx*(dndpt+dndpn+dndpv))/Ic;
A87=  0;
A88=(Ixz*(drdrt+drdrv+drdrw)+Ixx*(dndrt+dndrv))/Ic;
%

% longitudinal plant augmented is  X=[u w q theta]'
Flonaug=[A11 A12 A13 A14;A21 A22 A23 A24;A31 A32 A33 A34;
   A41 A42 A43 A44];
Plonaug=poly(Flonaug);
Rlonaug=roots(Plonaug);

% lateral plant augmented is  X=[v p phi r psi]'
Flataug=[A55 A56 A57 A58 0;A65 A66 A67 A68 0;
   A75 A76 A77 A78 0;A85 A86 A87 A88 0;0 0 0 1 0];
Plataug=poly(Flataug);
Rlataug=roots(Plataug);

% coupled plant
Amat=[A11 A12 A13 A14 A15 A16 A17 A18 0;
   A21 A22 A23 A24 A25 A26 A27 A28 0;
   A31 A32 A33 A34 A35 A36 A37 A38 0;
   A41 A42 A43 A44 A45 A46 A47 A48 0;
   A51 A52 A53 A54 A55 A56 A57 A58 0;
   A61 A62 A63 A64 A65 A66 A67 A68 0;
   A71 A72 A73 A74 A75 A76 A77 A78 0;
   A81 A82 A83 A84 A85 A86 A87 A88 0;
   0 0 0 0 0 0 0 1 0];

Pcoup=poly(Amat);
Rcoup=roots(Pcoup);

%
B11= dxdb1m*db1mddele/m;
B12= dxdthetom*dthetomddelc/m;
```

```
B13= dxda1m*da1mddela/m;
B14= 0;
%
B21= dzdb1m*db1mddele/m;
B22= dzdthetom*dthetomddelc/m;
B23= 0;
B24= 0;
%
B31= dmdb1m*db1mddele/Iyy;
B32= dmdthetom*dthetomddelc/Iyy;
B33= dmda1m*da1mddela/Iyy;
B34= 0;
%
B41= 0;
B42= 0;
B43= 0;
B44= 0;
%
B51= dydb1m*db1mddele/m;
B52= dydthetom*dthetomddelc/m;
B53= dyda1m*da1mddela/m;
B54= ((dydphin*dphinddelp+dyddelv*ddelvddelp+dydthetot*dthetotddelp)/m);
%
B61= Izz*drdb1m*db1mddele/Ic;
B62= (Izz*drdthetom*dthetomddelc+Ixz*dndthetom*dthetomddelc)/Ic;
B63= Izz*drda1m*da1mddela/Ic;
B64=
((Izz*(drdphin*dphinddelp+drddelv*ddelvddelp+drdthetot*dthetotddelp)+...

Ixz*(dndphin*dphinddelp+dnddelv*ddelvddelp+dndthetot*dthetotddelp))/Ic);
%
B71= 0;
B72= 0;
B73= 0;
B74= 0;
%
B81= Ixz*drdb1m*db1mddele/Ic;
B82= (Ixz*drdthetom*dthetomddelc+Ixx*dndthetom*dthetomddelc)/Ic;
B83= Ixz*drda1m*da1mddela/Ic;
B84=
((Ixz*(drdphin*dphinddelp+drddelv*ddelvddelp+drdthetot*dthetotddelp)+...

Ixx*(dndphin*dphinddelp+dnddelv*ddelvddelp+dndthetot*dthetotddelp))/Ic);
%
Glonaug=[B11 B12 B13 B14;
   B21 B22 B23 B24;
   B31 B32 B33 B34;
   B41 B42 B43 B44];
Glataug=[B51 B52 B53 B54;
   B61 B62 B63 B64;
   B71 B72 B73 B74;
   B81 B82 B83 B84;
   0 0 0 0];
% coupled input matrix
Bmat=[B11 B12 B13 B14;B21 B22 B23 B24;
   B31 B32 B33 B34;B41 B42 B43 B44;
   B51 B52 B53 B54;B61 B62 B63 B64;
```

```
   B71 B72 B73 B74;B81 B82 B83 B84;0 0 0 0];
%
xcouple=12/lockno*e/R/(1+e/3/R);
% designed damping
desdmdq=dmdqm+dmdqh+dmdqw;
desdrdp=drdpm+drdpt+drdpw+drdpv+drdpn;
desdndr=dndrm+dndrt+dndrv+dndrn;
% cooper harper pilot rating
prpitch=desdmdq/Iyy;
prroll=desdrdp/Ixx;
pryaw=desdndr/Izz;
% control power
cppitch=B31*Iyy;
cproll=B63*Ixx;
cpyaw=B84*Izz;
cpipitch=B31;
cpiroll=B63;
cpiyaw=B84;
%

save mat_temp Amat Bmat
stab_out_1
```

6. Ctrgrp.m

```
% Ctrgrp.m
% CALLED BY Cruise.m
% Computes the stability derivatives of the tail rotor in cruise flight.
% Computes the stability derivatives of the NOTAR in cruise flight.
% Uses data loaded in the workspace by JANRAD.M and STAB.M.
%
% Modified for JANRAD version 6.0 by LT David A. Heathorn
%
if tailrot==1

   % Compute the stability derivatives of the tail rotor in cruise
flight.

   dlampdydott=-1/((ohmt*Rt)*(1+dctsigdlampt*sigmat/(2*mut)));
   dydxdott=rho*Abt*(ohmt*Rt)^2*dctsigdmut*dmudxdot;
   dydydott=rho*Abt*(ohmt*Rt)^2*dctsigdlampt*dlampdydott;
   dydpt=dydydott*ht;
   dydrt=-dydydott*lt;
   dydthetot=rho*Abt*(ohmt*Rt)^2*dctsigdthetot;
   drdxdott=dydxdott*ht;
   drdydott=dydydott*ht;
   drdpt=dydpt*ht;
   drdrt=dydrt*ht;
   drdthetot=dydthetot*ht;
   dndxdott=-dydxdott*lt;
   dndydott=-dydydott*lt;
   dndpt=-dydpt*lt;
   dndrt=-dydrt*lt;
   dndthetot=dydthetot*lt*(-1);
   %
   %  If tail rotor is to be used, zero out NOTAR derivatives:
   %
   dydxdotn=0;dydydotn=0;dydzdotn=0;dydpn=0;dydrn=0;drdxdotn=0;
   drdydotn=0;drdzdotn=0;drdpn=0;drdrn=0;dndxdotn=0;dndydotn=0;
   dndzdotn=0;dndpn=0;dndrn=0;
   dndphin=0;drdphin=0;dydphin=0;
   dnddelv=0;drddelv=0;dyddelv=0;

elseif tailrot==3

   %
   % Compute the stability derivatives of the NOTAR in cruise flight.
   %
   qvin=.5*rho*v1max^2;
   dydxdotn=0;
   dydcsoar=qvin*bn^2*max([(-V/125+1),0]);
   % downwash blown off circulation control section above 125 ft/sec (75
kts)
   dcsoardcloar=1;
   dcloardpslot=2/pi;
   dpslotdcmu=dian/(2*bn)*sqrt(cmun*dian/(2*hslot))*1.5;
   dcmudvimax=-64*hslot/(dian*v1max);
   dvimaxdzdot=1;
   dcsoardswirl=-1/pi;
   dswirldydot=1/v1;
```

```
   dydydotn=dydcsoar*dcsoardswirl*dswirldydot;

dydzdotn=dydcsoar*dcsoardcloar*dcloardpslot*dpslotdcmu*dcmudvimax*dvimax
dzdot;
   dydpn=dydydotn*hn;
   dydrn=-dydydotn*ln;
   drdxdotn=dydxdotn*hn;
   drdydotn=dydydotn*hn;
   drdzdotn=dydzdotn*ln;
   drdpn=dydpn*hn;
   drdrn=dydrn*hn;
   dndxdotn=-dydxdotn*ln;
   dndydotn=-dydydotn*ln;
   dndzdotn=-dydzdotn*ln;
   dndpn=-dydpn*ln;
   dndrn=-dydrn*ln;
   dydphin=Ytmaxn/(pi/2);
   drdphin=dydphin*htn;
   dndphin=-dydphin*ltn;
   %
   %  If NOTAR is to be used, zero out tail rotor derivatives:
   %
   dydxdott=0;dydydott=0;dydpt=0;dydrt=0;dydthetot=0;drdxdott=0;
   drdydott=0;drdpt=0;drdrt=0;drdthetot=0;dndxdott=0;dndydott=0;
   dndpt=0;dndrt=0;dndthetot=0;
   mut=0;lampt=0;aot=0;b1st=0;a1st=0;locknot=0;
   drdthetot=0;dndthetot=0;dydthetot=0;
else
   disp(' ')
   disp('                              ERROR IN CTRGRP.M')
   disp('                   NO TAIL ROTOR OR NOTAR INSTALLED')
   disp('                              CHECK INPUT DATA')
   disp(' ')
   disp(' Press any key to continue...')
   pause
end
%
% return to CRUISE.M
```

7. Dctplots.m

```
% Dctplots.m
%
% plots ct/sigma, ch/sigma, cq/sigma, a1s and b1s
%                   VERSES
% mu, thetao and lamp
% fits a polynomial to the curve, takes a derivative of the curves
% then evaluates the appropriate parameters for CRUISE.M
%
% d(ct/sigma), d(ch/sigma), d(cq/sigma), da1s and db1s
%                   VERSES
% dmu, dthetao and dlamp
% uses the pertubation points from CRANK.M about nominal
%
% polyfitting data to obtain plots for n=1

% Modified for JANRAD version 6.0 by LT. David A. Heathorn

n=1;
ectsigmu=polyfit(vmu_mu,vctsig_mu,n);
ectsigtheto=polyfit(vthetao_to,vctsig_to,n);
ectsiglamp=polyfit(vlamp_la,vctsig_la,n);
%
echsigmu=polyfit(vmu_mu,vchsig_mu,n);
echsigtheto=polyfit(vthetao_to,vchsig_to,n);
echsiglamp=polyfit(vlamp_la,vchsig_la,n);
%
ecqsigmu=polyfit(vmu_mu,vcqsig_mu,n);
ecqsigtheto=polyfit(vthetao_to,vcqsig_to,n);
ecqsiglamp=polyfit(vlamp_la,vcqsig_la,n);
%
 ea1smu=polyfit(vmu_mu,va1s_mu,n);
 ea1stheto=polyfit(vthetao_to,va1s_to,n);
 ea1slamp=polyfit(vlamp_la,va1s_la,n);
%
 eb1smu=polyfit(vmu_mu,vb1s_mu,n);
 eb1stheto=polyfit(vthetao_to,vb1s_to,n);
 eb1slamp=polyfit(vlamp_la,vb1s_la,n);
%
%taking derivatives
%
edctsigdmu=deriv1(ectsigmu);
edctsigdtheto=deriv1(ectsigtheto);
edctsigdlamp=deriv1(ectsiglamp);
%
edchsigdmu=deriv1(echsigmu);
edchsigdtheto=deriv1(echsigtheto);
edchsigdlamp=deriv1(echsiglamp);
%
edcqsigdmu=deriv1(ecqsigmu);
edcqsigdtheto=deriv1(ecqsigtheto);
edcqsigdlamp=deriv1(ecqsiglamp);
%
eda1sdmu=deriv1(ea1smu);
eda1sdtheto=deriv1(ea1stheto);
eda1sdlamp=deriv1(ea1slamp);
```

```
%
edb1sdmu=deriv1(eb1smu);
edb1sdtheto=deriv1(eb1stheto);
edb1sdlamp=deriv1(eb1slamp);
%
% evaluating derivatives
%
dctsigdmu=polyval(edctsigdmu,vmu_mu(1,2))
dctsigdtheto=polyval(edctsigdtheto,vthetao_to(1,2))
dctsigdlamp=polyval(edctsigdlamp,vlamp_la(1,2))
%
dchsigdmu=polyval(edchsigdmu,vmu_mu(1,2))
dchsigdtheto=polyval(edchsigdtheto,vthetao_to(1,2))
dchsigdlamp=polyval(edchsigdlamp,vlamp_la(1,2))
%
dcqsigdmu=polyval(edcqsigdmu,vmu_mu(1,2))
dcqsigdtheto=polyval(edcqsigdtheto,vthetao_to(1,2))
dcqsigdlamp=polyval(edcqsigdlamp,vlamp_la(1,2))
%
da1dmu=polyval(eda1sdmu,vmu_mu(1,2));
da1dtheto=polyval(eda1sdtheto,vthetao_to(1,2));

% Due to poor modeling of flapping find da1dlamp by diferintiating
equation (8) from
% Gessow and Myers "Aerodynamics of Helicopters" pg 186 eq(8) and assume
the change
% in flapping is the same as the change in the tip path plane.

a1num1=(1/vmu_mu(1,2));
a1num2=(vctsig_mu(1,2)/solidity/-
4/vmu_mu(1,2))*(vmu_mu(1,2)^2+vlamp_la(1,2)^2)^(-3/2);
a1num2=2*-vlamp_la(1,2)*a1num2;
a1num3=(solidity*dctsigdlamp/2/vmu_mu(1,2))*(vmu_mu(1,2)^2+vlamp_la(1,2)
^2)^(-1/2);
da1dlamp=a1num1+a1num2+a1num3;


db1dmu=polyval(edb1sdmu,vmu_mu(1,2));
db1dtheto=polyval(edb1sdtheto,vthetao_to(1,2));
db1dlamp=polyval(edb1sdlamp,vlamp_la(1,2));
```

8. Dctplott.m

```
% Dctplott.m
%
% plots ct/sigmat
%      VERSES
% mut, thetaot and lampt
% fits a polynomial to the curve, takes a derivative of the curves
% then evaluates the appropriate parameters for CRUISE.M
%
% d(ct/sigma)
%          VERSES
% dmut, dthetaot and dlampt
% uses the pertubation points from JANRAD about nominal
%
% polyfitting data to obtain plots for n=1

% Modified for JANRAD version 6.0 by David A. Heathorn

n=1;
ectsigmut=polyfit(vmut,vctsigt_mut,n);
ectsigthetot=polyfit(vthetaot,vctsigt_tot,n);
ectsiglampt=polyfit(vlampt,vctsigt_lat,n);
%
%taking derivatives
%
edctsigdmut=deriv1(ectsigmut);
edctsigdthetot=deriv1(ectsigthetot);
edctsigdlampt=deriv1(ectsiglampt);
%
% evaluating derivatives
%
dctsigdmut=polyval(edctsigdmu,vmut(1,2));
dctsigdthetot=polyval(edctsigdtheto,vthetaot(1,2));
dctsigdlampt=polyval(edctsigdlamp,vlampt(1,2));
```

9. Deriv1.m

```
% DERIV1.M
% computes the derivative of a polynomial using first order curve
%
% Created for JANRAD version 6.0 by LT David A. Heathorn

function[deriv]=deriv1(polyno);
der=[1 0];
d=der.*polyno;
deriv=d(1,1);
```

10. Eigen_plotter.m

```
% Eigen_ploter.m

% M-file called by time_freq_resp_fcn.m to plot eigen values of linear
model.

% Created for JANRAD version 6.0 by LT David A. Heathorn

global Amat Bmat C u


figure
plot(real(eig(Amat(1:8,1:8))),imag(eig(Amat(1:8,1:8))),'kX','markersize'
,12)
title('Eigen Values of Coupled Matrix A')
xlabel('Real Axis')
ylabel('Imag Axis')
grid



figure
plot(real(eig(Amat(1:4,1:4))),imag(eig(Amat(1:4,1:4))),'kX','markersize'
,12)
xlabel('Real Axis')
ylabel('Imag Axis')
title('Eigen Values of Uncoupled Longitudinal Matrix A')
grid

figure
plot(real(eig(Amat(5:8,5:8))),imag(eig(Amat(5:8,5:8))),'kX','markersize'
,12)
xlabel('Real Axis')
ylabel('Imag Axis')
title('Eigen Values of Uncoupled Lateral Matrix A')
grid
```

11. Hmrgrp.m

```
% Hmrgrp.m
% CALLED BY Hover.m
% Computes the basic main rotor derivatives at a hover
% Computes the stability derivatives of the main rotor at a hover
% Uses data loaded in the workspace by JANRAD.M and STAB.M
%
% Compute the basic mainrotor derivatives at a hover
%
% Modified for JANRAD version 6.0 by LT David A. Heathorn
%
dmudxdot=1/ohm/R;
dlampdzdot=dmudxdot;
da1dmu=8/3*thetao+2*theta1-2*v1/ohm/R;
db1dmu=4/3*ao;
dctsigdlamp=inv(8/a+(sqrt(sigma/2)/(sqrt(ctsig))));
dcqsigdlamp=-a/4*(theta75-2*v1/ohm/R);
da1dq=-16/(lockno*ohm*(1-e/R)^2)-12*e/R/(lockno*ohm*(1-e/R)^3);
db1dp=da1dq;
da1dp=1/ohm*(1-(192*e/R/(lockno^2*(1-e/R)^5)));
db1dq=-da1dp;
da1da=12*e/R/(lockno*(1-e/R)^3);
db1db=da1da;
da1db=-1/(1+((144*(e/R)^2)/(lockno^2*(1-e/R)^6)));
db1da=-da1db;
dchsigda=3/2*ctsig*(1-a/18*theta75/ctsig);
dcysigdb=dchsigda;
dmda1s=3/4*e/R*Ab*rho*R*(ohm*R)^2*a/lockno;
drdb1s=dmda1s;
%
% Compute the mainrotor stability derivatives at a hover
%
dxdxdotm=-rho*Ab*(ohm*R)^2*dchsigda*da1dmu*dmudxdot;
dxdydotm=-rho*Ab*(ohm*R)^2*dcysigdb*db1dmu*dmudxdot;
dxdzdotm= 0;
dxdqm=-rho*Ab*(ohm*R)^2*dchsigda*da1dq;
dxdpm=-rho*Ab*(ohm*R)^2*dchsigda*da1dp ;
dxdrm=0;
dxdthetom=-rho*Ab*(ohm*R)^2*(a1s+im)*dctsigdtheto;
dxda1m=-rho*Ab*(ohm*R)^2*dchsigda*da1da;
dxdb1m=-rho*Ab*(ohm*R)^2*dchsigda*da1db;
dydxdotm=rho*Ab*(ohm*R)^2*dcysigdb*db1dmu*dmudxdot;
dydydotm=dxdxdotm;
dydzdotm=0;
dydqm=dxdpm;
dydrm=0;
dydpm=-dxdqm;
dydthetom=rho*Ab*(ohm*R)^2*(2*b1s)*dctsigdtheto;
dyda1m=dxdb1m;
dydb1m=-dxda1m;
dzdxdotm=0;
dzdydotm=0;
dzdzdotm=-rho*Ab*(ohm*R)^2*dctsigdlamp*dlampdzdot;
dzdpm=0;
dzdqm=0;
dzdrm=0;
```

```
dzdthetom=-rho*Ab*(ohm*R)^2*dctsigdtheto;
dzdb1m=0;
drdxdotm=drdb1s*db1dmu*dmudxdot+dydxdotm*hm+dzdxdotm*hm;
drdydotm=-dmda1s*da1dmu*dmudxdot+dxdxdotm*hm;
drdzdotm=dzdzdotm*ym;
drdqm=drdb1s*db1dq+dydqm*hm;
drdrm=0;
drdpm=drdb1s*db1dp+dydpm*hm;
drdthetom=dydthetom*hm+dzdthetom*ym;
drda1m=drdb1s*db1da+dyda1m*hm;
drdb1m=drdb1s*db1db+dydb1m*hm;
dmdxdotm=dmda1s*da1dmu*dmudxdot-dxdxdotm*hm;
dmdydotm=drdxdotm;
dmdzdotm=dzdzdotm*lm;
dmdqm=dmda1s*da1dq-dxdqm*hm;
dmdrm=0;
dmdpm=dmda1s*da1dp-dxdpm*hm;
dmdthetom=-dxdthetom*hm+dzdthetom*lm;
dmda1m=dmda1s*da1da-dxda1m*hm;
dmdb1m=dmda1s*da1db-dxdb1m*hm;
dndxdotm=0;
dndzdotm=rho*Ab*(ohm*R)^2*R*dcqsigdlamp*dlampdzdot;
dndydotm=0;
dndqm=0;
dndpm=0;
dndrm=-2*rho*Ab*(ohm*R)^2*R*cqsig;
dndthetom=rho*Ab*(ohm*R)^2*R*dcqsigdtheto;
%
% return to HOVER.M
```

12. Hover.m

```
% Hover.m
% CALLED BY STAB.M
% Computes the stability derivatives at a hover.
% calls the FOLLOWING subroutines to compute stability derivatives
%
% HMRGRP
% HTRGRP
% TRIM
%
% computation of stability derivatives
% the only derivatives important at hover are main and tail rotor
format compact
% evaluate dctsigdtheto dcqsigdtheto dctsigdthetot and dcqsigdthetot
%
% Modified for JANRAD version 6.0 by LT David A. Heathorn

global S_SC_INPUT_1 S_SC_INPUT_2 S_PERF_INPUT S_PERF_OUTPUT H_S_SC_STAT
S_H_SC_STAT Amat Bmat

% Unstructure input variables

S_STAB_INPUT_1=S_SC_INPUT_1;
S_STAB_INPUT_2=S_SC_INPUT_2;

unstructure
unstructure_stab_input_1
unstructure_stab_input_2

H_SC_STAT=S_H_SC_STAT.h_sc_stat;
H_STATUS=S_H_SC_STAT.h_status;
H_STATUS1=S_H_SC_STAT.h_status1;
H_STATUS2=S_H_SC_STAT.h_status2;
H_STATUS3=S_H_SC_STAT.h_status3;
H_STATUS4=S_H_SC_STAT.h_status4;
H_BK=S_H_SC_STAT.h_bk;
H_GO=S_H_SC_STAT.h_go;
H_RUPT=S_H_SC_STAT.h_rupt;
H_RES=S_H_SC_STAT.h_res;
H_MEN=S_H_SC_STAT.h_men;


% Turn Stab_on flag on

STAB_ON=1;

% VARIABLE UNIT CONVERSION

im=im/57.3;
delta3=delta3/57.3;
theta1t=theta1t/57.3;
alplov=alplov/57.3;
alploh=alploh/57.3;
ih=ih/57.3;
alplow=alplow/57.3;
```

```
iw=iw/57.3;
swirl=swirl/57.3;
db1mddele=db1mddele/57.3;
da1mddela=da1mddela/57.3;
dthetomddelc=dthetomddelc/57.3;
dthetotddelp=dthetotddelp/57.3;
sidearm=sidearm*2;
if sidearm==0
   sidearm=1e3;
end
dphinddelp=pi/sidearm;
sidearm=pi/(sidearm*2);


vctsig=zeros(1,2);
vcqsig=zeros(1,2);
vthetao=zeros(1,2);
vctsigt=ones(1,2);
vcqsigt=ones(1,2);
vthetaot=ones(1,2);

theta1=-twist;
GW=GW*1.0005;   % .5% HIGHER
set(H_STATUS2,'STRING', 'HOVER ROUTINE')
set(H_STATUS4,'STRING', 'FIRST OF TWO PERTURBATIONS')
tic
set(H_STATUS1,'STRING', 'START ELAPSED TIME')
pause(3)
Trim
%   *** Calculation of output parameters from trim ***
Qrotor=mean(DMpsi)*b;
solidity=b*(sum(cblade)/length(cblade))/(pi*R);
CQ=Qrotor/(Adisk*rho*Vtip^2*R);
CH=Hrotor/(Adisk*rho*Vtip^2);
CH_sig=CH/solidity;
vctsig(1,2)=CT/solidity;
vcqsig(1,2)=CQ/solidity;
thetasave=thetao;
vthetao(1,2)=thetasave-.7*theta1;
ohm=omega;
sigma=solidity;
g=32.174;
A=pi*R*R;
if tailrot==1
   Abt=bt*Rt*ct;
   At=pi*Rt*Rt;
   mut=Vinf/ohmt/Rt;
   sigmat=Abt/At;
   lt=ltd-xcg;
   ht=htd-zcg;
   yt=ytd-ycg;
   locknot=rho*at*ct*Rt^4/Ibt;
   Tt=Qrotor/lt;
   ctsigt=Tt/(rho*At*(ohmt*Rt)^2);
   v1t=sqrt(Tt/2/rho/At);
   lampt=-v1t/ohmt/Rt;
   aot=2/3*locknot*ctsigt/at-3/2*g*Rt^2/(ohmt*Rt)^2;
```

```
   thetaot=3/2*(4/at*ctsigt+sqrt(CQ/2))-3/4*theta1t;
   a1st=0;
   b1st=0;
   theta75t=thetaot+.75*theta1t;
   v1t=sqrt(Tt/2/rho/At);
   aoat=theta75t+atan(v1t/ohmt/Rt/.7);
   cdt=.006+(aoat*at)^2/pi/Rt/ct;
   phit=sqrt(ctsigt*sigmat/2);
   thet=4/at*ctsigt+sqrt(phit);
   cqsigt=sigmat/2*(at/2*(thet-phit)*phit+cdt/4);


   vctsigt(1,2)=ctsigt;
   vcqsigt(1,2)=cqsigt;
   vthetaot(1,2)=thetaot;
end

save stabtemp vctsig vcqsig vthetao theta1 vctsigt vcqsigt vthetaot g...
S_SC_INPUT_1 S_SC_INPUT_2 S_PERF_INPUT S_PERF_OUTPUT H_S_SC_STAT
S_H_SC_STAT

clear

load stabtemp

S_STAB_INPUT_1=S_SC_INPUT_1;
S_STAB_INPUT_2=S_SC_INPUT_2;

unstructure
unstructure_stab_input_1
unstructure_stab_input_2

H_SC_STAT=S_H_SC_STAT.h_sc_stat;
H_STATUS=S_H_SC_STAT.h_status;
H_STATUS1=S_H_SC_STAT.h_status1;
H_STATUS2=S_H_SC_STAT.h_status2;
H_STATUS3=S_H_SC_STAT.h_status3;
H_STATUS4=S_H_SC_STAT.h_status4;
H_BK=S_H_SC_STAT.h_bk;
H_GO=S_H_SC_STAT.h_go;
H_RUPT=S_H_SC_STAT.h_rupt;
H_RES=S_H_SC_STAT.h_res;
H_MEN=S_H_SC_STAT.h_men;


% Turn Stab_on flag on

STAB_ON=1;

% VARIABLE UNIT CONVERSION

im=im/57.3;
delta3=delta3/57.3;
theta1t=theta1t/57.3;
alplov=alplov/57.3;
alploh=alploh/57.3;
ih=ih/57.3;
```

```
alplow=alplow/57.3;
iw=iw/57.3;
swirl=swirl/57.3;
db1mddele=db1mddele/57.3;
da1mddela=da1mddela/57.3;
dthetomddelc=dthetomddelc/57.3;
dthetotddelp=dthetotddelp/57.3;
sidearm=sidearm*2;
if sidearm==0
   sidearm=1e3;
end
dphinddelp=pi/sidearm;
sidearm=pi/(sidearm*2);

set(H_STATUS4,'STRING', 'SECOND OF TWO TRIM PERTURBATIONS')
set(H_STATUS1,'STRING',['RUN ELAPSED TIME IS ',fix(num2str(toc)),'
SECONDS'])
pause(3)
Trim
set(H_STATUS4,'STRING', 'PERTURBATIONS COMPLETE - EVALUATING STABILITY
DERIVATIVES')
set(H_STATUS1,'STRING',['RUN ELAPSED TIME IS ',fix(num2str(toc)),'
SECONDS'])
pause(3)

Qrotor=mean(DMpsi)*b;
solidity=b*(sum(cblade)/length(cblade))/(pi*R);
CQ=Qrotor/(Adisk*rho*Vtip^2*R);
CH=Hrotor/(Adisk*rho*Vtip^2);
vctsig(1,1)=CT/solidity;
vcqsig(1,1)=CQ/solidity;
thetasave=thetao;
vthetao(1,1)=thetasave-.7*theta1;
dctsigdtheto=(vctsig(1,1)-vctsig(1,2))/(vthetao(1,1)-vthetao(1,2));
dcqsigdtheto=(vcqsig(1,1)-vcqsig(1,2))/(vthetao(1,1)-vthetao(1,2));
ohm=omega;
sigma=solidity;
A=pi*R*R;
if tailrot==1
   Abt=bt*Rt*ct;
   At=pi*Rt*Rt;
   mut=Vinf/ohmt/Rt;
   sigmat=Abt/At;
   lt=ltd-xcg;
   ht=htd-zcg;
   yt=ytd-ycg;
   locknot=rho*at*ct*Rt^4/Ibt;
   Tt=Qrotor/lt;
   ctsigt=Tt/(rho*At*(ohmt*Rt)^2);
   v1t=sqrt(Tt/2/rho/At);
   lampt=-v1t/ohmt/Rt;
   aot=2/3*locknot*ctsigt/at-3/2*g*Rt^2/(ohmt*Rt)^2;
   thetaot=3/2*(4/at*ctsigt+sqrt(CQ/2))-3/4*theta1t;
   a1st=0;
   b1st=0;
   theta75t=thetaot+.75*theta1t;
   v1t=sqrt(Tt/2/rho/At);
```

```
   aoat=theta75t+atan(v1t/ohmt/Rt/.7);
   cdt=.006+(aoat*at)^2/pi/Rt/ct;
   phit=sqrt(ctsigt*sigmat/2);
   thet=4/at*ctsigt+sqrt(phit);
   cqsigt=sigmat/2*(at/2*(thet-phit)*phit+cdt/4);
   vctsigt(1,1)=ctsigt;
   vcqsigt(1,1)=cqsigt;
   vthetaot(1,1)=thetaot;
   dctsigdthetot=(vctsigt(1,2)-vctsigt(1,1))/(vthetaot(1,2)-
vthetaot(1,1));
   dcqsigdthetot=(vcqsigt(1,2)-vcqsigt(1,1))/(vthetaot(1,2)-
vthetaot(1,1));
end

% CONFIGURATION CALCULATIONS

theta7=thetao;
thetao=theta7-.7*theta1;
ctsig=CT/solidity;
chsig=CH/solidity;
cqsig=CQ/solidity;
altpp=-alphaT;
A1=-theta1c;
B1=-theta1s;
ao=betao;
v1=mean(vi);
V=Vinf;
% Main rotor
g=32.2;
hm=hmd-zcg;
ym=ymd-ycg;
lm=lmd-xcg;
c=(sum(cblade)/length(cblade));
Ab=c*R*4;
lamp=0;
mu=0;
ct=CT;
f=Afv;
thetat=4/a*ctsig+sqrt(sigma*ctsig/2);
lockno=rho*a*c*R^4/Ib;
theta75=thetao+.75*theta1;
num1=thetao*(8/3+32/45*mu^3/pi);
num2=theta1*(2+mu^4/12);
num3=lamp*(2-mu*mu/2);
den1=1+3/2*mu*mu-5*mu^4/24;
a1s=mu*(num1+num2+num3)/den1-B1;
b1s=a1s*12/lockno*e/R/(1+e/R/3);
m=GW/g;
Ic=Ixx*Izz;
if tailrot==3;
   % NOTAR
   v1max=1.2*v1;      %  downwash at the boom slots (NOTAR)
   delvmax=pi/4;
   if maxr==0
      ddelvddelp=0;
   else
      ddelvddelp=delvmax/maxr;   % verticle fin deflection/maxr
```

```
    end
    ltn=lttnd-xcg;
    ln=ltnd-xcg;
    yn=ytnd-xcg;
    hn=htnd-zcg;
    htn=hn;
    Tt=cqsig*sigma*rho*A*(ohm*R)^2*R/ltn;
    cmun=.55;    % typical value
    bn=.5*R;
    hslot=.028*dian/2;
    arn=bn/dian;
    dthetoddelp = 0;
end
% trim conditions
tho=-T*(im+a1s)/GW;
pho=-(T*b1s+Tt)/GW;
wo=0;
vo=0;
uo=0;

% STABILITY CALCULATION
%
Hmrgrp
Htrgrp
%
% computation of A,B,C,D matrices
%
A11= (dxdxdotm)/m;
A12= (dxdzdotm)/m;
A13= (dxdqm)/m-wo;
A14= -g*cos(tho);
A15= (dxdydotm)/m;
A16= (dxdpm)/m;
A17= 0;
A18= (dxdrm)/m+vo;
%
A21= (dzdxdotm)/m;
A22= (dzdzdotm)/m;
A23= (dzdqm)/m+uo;
A24= -g*cos(pho)*sin(tho);
A25= (dzdydotm+dzdydott)/m;
A26= (dzdpm)/m-vo;
A27= -g*sin(pho)*cos(tho);
A28= (dzdrm)/m;
%
A31= (dmdxdotm)/Iyy;
A32= (dmdzdotm)/Iyy;
A33= (dmdqm)/Iyy;
A34= 0;
A35= (dmdydotm+dmdydott+dmdydotn)/Iyy;
A36= (dmdpm)/Iyy;
A37= 0;
A38= (dmdrm+dmdrt+dmdrn)/Iyy;
%
A41= 0;
A42= 0;
A43= cos(pho);
```

```
A44= 0;
A45= 0;
A46= 0;
A47= 0;
A48= -sin(pho);
%
A51= (dydxdotm)/m;
A52= (dydzdotm+dydzdotn)/m;
A53= (dydqm)/m;
A54= -g*sin(pho)*sin(tho);
A55= (dydydotm+dydydott+dydydotn)/m;
A56= (dydpm+dydpt+dydpn)/m+wo;
A57= g*cos(pho)*cos(tho);
A58= (dydrm+dydrt+dydrn)/m-uo;
%
A61= (Izz*(drdxdotm+drdxdotn)+Ixz*(dndxdotm+dndxdotn))/Ic;
A62= (Izz*(drdzdotm+drdzdotn)+Ixz*(dndzdotm+dndzdotn))/Ic;
A63= (Izz*(drdqm)+Ixz*(dndqm))/Ic;
A64= 0;
A65=
(Izz*(drdydotm+drdydott+drdydotn)+Ixz*(dndydotm+dndydott+drdydotn))/Ic;
A66= (Izz*(drdpm+drdpt+drdpn)+Ixz*(dndpm+dndpt+dndpn))/Ic;
A67= 0;
A68= (Izz*(drdrm+drdrt+drdrn)+Ixz*(dndrm+dndrt+dndrn))/Ic;
%
A71= 0;
A72= 0;
A73= sin(pho)*tan(tho);
A74= 0;
A75= 0;
A76= 1;
A77= 0;
A78= cos(pho)*tan(tho);
%
A81= (Ixz*(drdxdotm+drdxdotn)+Ixx*(dndxdotm+dndxdotn))/Ic;
A82= (Ixz*(drdzdotm+drdzdotn)+Ixx*(dndzdotm+dndzdotn))/Ic;
A83= (Ixz*(drdqm)+Ixx*(dndqm))/Ic;
A84= 0;
A85=
(Ixz*(drdydotm+drdydott+drdydotn)+Ixx*(dndydotm+dndydott+dndydotn))/Ic;
A86= (Ixz*(drdpm+drdpt+drdpn)+Ixx*(dndpm+dndpt+dndpn))/Ic;
A87=  0;
A88= (Ixz*(drdrm+drdrt+drdrn)+Ixx*(dndrm+dndrt+dndrn))/Ic;
%
% longitudinal plant augmented X=[u w q theta]
Flonaug=[A11 A12 A13 A14;A21 A22 A23 A24;A31 A32 A33 A34;
   A41 A42 A43 A44];
Plonaug=poly(Flonaug);
Rlonaug=roots(Plonaug);
%
% Lateral plant augmented with X=[v p phi r psi]'
Flataug=[A55 A56 A57 A58 0;A65 A66 A67 A68 0;
   A75 A76 A77 A78 0;A85 A86 A87 A88 0;0 0 0 1 0];
Plataug=poly(Flataug);
Rlataug=roots(Plataug);
% coupled plant
Amat=[A11 A12 A13 A14 A15 A16 A17 A18 0;
```

```
   A21 A22 A23 A24 A25 A26 A27 A28 0;
   A31 A32 A33 A34 A35 A36 A37 A38 0;
   A41 A42 A43 A44 A45 A46 A47 A48 0;
   A51 A52 A53 A54 A55 A56 A57 A58 0;
   A61 A62 A63 A64 A65 A66 A67 A68 0;
   A71 A72 A73 A74 A75 A76 A77 A78 0;
   A81 A82 A83 A84 A85 A86 A87 A88 0;
   0 0 0 0 0 0 0 1 0];
Pcoup=poly(Amat);
Rcoup=roots(Pcoup);
%
B11= dxdb1m*db1mddele/m;
B12= dxdthetom*dthetomddelc/m;
B13= dxda1m*da1mddela/m;
B14= 0;
%
B21= dzdb1m*db1mddele/m;
B22= dzdthetom*dthetomddelc/m;
B23= 0;
B24= 0;
%
B31= dmdb1m*db1mddele/Iyy;
B32= dmdthetom*dthetomddelc/Iyy;
B33= dmda1m*da1mddela/Iyy;
B34= (dmdphin*dphinddelp+dmdthetot*dthetotddelp)/Iyy;
%
B41= 0;
B42= 0;
B43= 0;
B44= 0;
%
B51= dydb1m*db1mddele/m;
B52= dydthetom*dthetomddelc/m;
B53= dyda1m*da1mddela/m;
B54= (dydphin*dphinddelp+dydthetot*dthetotddelp)/m;
%
B61= Izz*drdb1m*db1mddele/Ic;
B62= (Izz*drdthetom*dthetomddelc+Ixz*dndthetom*dthetomddelc)/Ic;
B63= Izz*drda1m*da1mddela/Ic;
B64= (Izz*(drdphin*dphinddelp+drdthetot*dthetotddelp)+...
   Ixz*(dndphin*dphinddelp+dndthetot*dthetotddelp))/Ic;
%
B71= 0;
B72= 0;
B73= 0;
B74= 0;
%
B81= Ixz*drdb1m*db1mddele/Ic;
B82= (Ixz*drdthetom*dthetomddelc+Ixx*dndthetom*dthetomddelc)/Ic;
B83= Ixz*drda1m*da1mddela/Ic;
B84=  (Ixz*(drdphin*dphinddelp+drdthetot*dthetotddelp)+...
   Ixx*(dndphin*dphinddelp+dndthetot*dthetotddelp))/Ic;
%
Glonaug=[B11 B12 B13 B14;
   B21 B22 B23 B24;
   B31 B32 B33 B34;
   B41 B42 B43 B44];
```

```
%
Glataug=[B51 B52 B53 B54;
   B61 B62 B63 B64;
   B71 B72 B73 B74;
   B81 B82 B83 B84;
   0 0 0 0];
% coupled input matrix
Bmat=[B11 B12 B13 B14;B21 B22 B23 B24;
   B31 B32 B33 B34;B41 B42 B43 B44;
   B51 B52 B53 B54;B61 B62 B63 B64;
   B71 B72 B73 B74;B81 B82 B83 B84;0 0 0 0];
%
% cross coupling
xcouple=12/lockno*e/R/(1+e/3/R);
% designed damping
desdmdq=dmdqm;
desdrdp=drdpm+drdpt+drdpn;
desdndr=dndrm+dndrt+dndrn;
% now cooper harper pilot rating
prpitch=desdmdq/Iyy;
prroll=(drdpm+drdpt+drdpn)/Ixx;
pryaw=desdndr/Izz;
% control power
cppitch=B31*Iyy;
cproll=B63*Ixx;
cpyaw=B84*Izz;
cpipitch=B31;
cpiroll=B63;
cpiyaw=B84;
%
thetao=theta7;

save mat_temp Amat Bmat

stab_out_1
```

13. Htrgrp.m

```
% Htrgrp.m
% CALLED BY Hover.m
% Computes the basic tail rotor or NOTAR derivatives at a hover
% Computes the stability derivatives of the tail rotor or NOTAR at a
hover
% OR
% Computes the stability derivatives of the NOTAR at a hover
%
% Uses data loaded in the workspace by JANRAD.M and STAB.M
%
% Modified for JANRAD version 6.0 by LT David A. Heathorn
%
if tailrot==1
   %
   % Compute the basic tail rotor derivatives at a hover.
   %
   dmudxdott=1/ohmt/Rt;
   dlampdydott=-dmudxdott;
   da1dmut=8/3*thetao+2*theta1-2*v1/ohmt/Rt;
   db1dmut=4/3*aot;
   dctsigdlampt=inv(8/at+(sqrt(sigmat/2)/(sqrt(ctsig))));
   dcqsigdlampt=-at/4*(theta75t-2*v1t/ohmt/Rt);
   da1dqt=0;
   db1dpt=0;
   da1dpt=0;
   db1dqt=0;
   da1dat=0;
   db1dbt=0;
   da1dbt=0;
   db1dat=0;
   dchsigdat=3/2*ctsig*(1-at/18*theta75t/ctsig);
   dcysigdbt=dchsigdat;
   dmda1st=0;
   drdb1st=0;
   %
   % Compute the tail rotor stabitity derivatives at a hover
   %
   dydydott=rho*Abt*(ohmt*Rt)^2*dctsigdlampt*dlampdydott;
   dydpt=dydydott*ht;
   dydrt=-dydydott*lt;
   dydthetot=rho*Abt*(ohmt*Rt)^2*dctsigdthetot;
   drdydott=dydydott*ht;
   drdrt=dydrt*ht;
   drdpt=dydpt*ht;
   drdthetot=dydthetot*ht;
   dmdydott=rho*Abt*(ohmt*Rt)^2*Rt*dcqsigdlampt*dlampdydott;
   dmdrt=-dmdydott*lt;
   dmdthetot=rho*Abt*(ohmt*Rt)^2*Rt*dcqsigdthetot;
   dndydott=-dydydott*lt;
   dndpt=-dydpt*lt;
   dndrt=-dydrt*lt;
   dndthetot=-dydthetot*lt;
   %
   % If tail rotor is used, zero out NOTAR derivatives:
   %
```

```
   dydxdotn=0;dydydotn=0;dydzdotn=0;dydpn=0;dydrn=0;
   dzdydott=0;
   drdxdotn=0;drdydotn=0;drdzdotn=0;drdpn=0;drdrn=0;
   dmdydotn=0;dmdrn=0;
   dndxdotn=0;dndydotn=0;dndzdotn=0;dndpn=0;dndrn=0;
   dmdphin=0;drdphin=0;dndphin=0;dydphin=0;
elseif tailrot==3

   % Compute the NOTAR derivatives at a hover.
   %
   qvin=.5*rho*v1max^2;
   dydxdotn=0;
   dydcsoar=qvin*bn^2;
   dcsoardcloar=1;
   dcloardpslot=2/pi;
   dpslotdcmu=dian/(2*bn)*sqrt(cmun*dian/(2*hslot))*1.5;
   dcmudvimax=-64*hslot/(dian*v1max);
   dvimaxdzdot=1;
   dcsoardswirl=-1/pi;
   dswirldydot=1/v1;
   dydydotn=dydcsoar*dcsoardswirl*dswirldydot;

dydzdotn=dydcsoar*dcsoardcloar*dcloardpslot*dpslotdcmu*dcmudvimax*dvimax
dzdot;
   dydpn=dydydotn*hn;
   dydrn=-dydydotn*ln;
   drdxdotn=dydxdotn*hn;
   drdydotn=dydydotn*hn;
   drdzdotn=dydzdotn*ln;
   drdpn=dydpn*hn;
   drdrn=dydrn*hn;
   dmdydotn=0;
   dmdrn=-dmdydotn*ln;
   dndxdotn=-dydxdotn*ln;
   dndydotn=-dydydotn*ln;
   dndzdotn=-dydzdotn*ln;
   dndpn=-dydpn*ln;
   dndrn=-dydrn*ln;
   dydphin=Ytmaxn/(pi/2);
   drdphin=dydphin*htn;
   dmdphin=0;
   dndphin=-dydphin*ltn;
   %
   %  If NOTAR is to be used, zero out tail rotor derivatives:
   %
   dydydott=0;dydpt=0;dydrt=0;dydthetot=0;drdydott=0;
   dzdydott=0;
   drdrt=0;drdpt=0;drdthetot=0;
   dmdydott=0;dmdrt=0;dmdthetot=0;
   dndydott=0;dndpt=0;dndrt=0;dndthetot=0;
   lampt=0;mut=0;aot=0;b1st=0;a1st=0;locknot=0;
   dmdthetot=0;dndthetot=0;drdthetot=0;dydthetot=0;
else
   disp(' ')
   disp('                              ERROR IN HTRGRP.M')
   disp('                    NO TAIL ROTOR OR NOTAR INSTALLED')
   disp('                              CHECK INPUT DATA')
```

```
   disp(' ')
   disp(' Press any key to continue...')
   pause
end
%
% return to HOVER.M
```

14. Imp_plotter.m

```
% Imp_plotter.m

% M-file called by time_freq_resp_fcn.m to plot the impulse response
% for specified input and output.

% Created for JANRAD version 6.0 by LT David A. Heathorn


global Amat Bmat u C T_STOP T_INC


D=0;
t_start=0;
t_inc=T_INC;
t_stop=T_STOP;

t=t_start:t_inc:t_stop;

if u(1)==1

   for j=1:length(C)
      if C(j,j)==1
         figure
         [num,den]=ss2tf(Amat,Bmat(:,1),C(j,:),D);
         [y]=impulse(num,den,t);
         plot(t,y)
         xlabel('time (sec)')
         grid
        if j==1
            title('Response of x-velocity (u) to Longitudinal Cyclic
Unit Impulse Input')
            ylabel('Velocity (ft/sec)')
         elseif j==2
            title('Response of z-velocity (w) to Longitudinal Cyclic
Unit Impulse Input')
            ylabel('Velocity (ft/sec)')
         elseif j==3
            title('Response of Pitch Rate (q) to Longitudinal Cyclic
Unit Impulse Input')
            ylabel('Pitch Rate (rad/sec)')
         elseif j==4
            title('Response of Pitch Angle (theta) to Longitudinal
Cyclic Unit Impulse Input')
            ylabel('Pitch Angle (rad.)')
         elseif j==5
            title('Response of y-veloctiy (v) to Longitudinal Cyclic
Unit Impulse Input')
            ylabel('Velocity (ft/sec)')
         elseif j==6
            title('Response of Roll Rate (p) to Longitudinal Cyclic Unit
Impulse Input')
            ylabel('Roll Rate (rad/sec)')
         elseif j==7
            title('Response of Roll Angle (phi) to Longitudinal Cyclic
Unit Impulse Input')
```

```
            ylabel('Roll Angle (rad)')
         elseif j==8
            title('Response of Yaw Rate (r) to Longitudinal Cyclic Unit
Impulse Input')
            ylabel('Yaw Rate (rad/sec)')
         elseif j==9
            title('Response of Yaw Angle (psi) to Longitudinal Cyclic
Unit Impulse Input')
            ylabel('Yaw Angle(rad)')
         end
      end
   end

elseif u(2)==1

   for j=1:length(C)
      if C(j,j)==1
         figure
         [num,den]=ss2tf(Amat,Bmat(:,1),C(j,:),D);
         [y]=impulse(num,den,t);
         plot(t,y)
         xlabel('time (sec)')
         grid
         xlabel('time (sec)')
         if j==1
            title('Response of x-velocity (u) to Collective Unit Impulse
Input')
            ylabel('Velocity (ft/sec)')
         elseif j==2
            title('Response of z-velocity (w) to Collective Unit Impulse
Input')
            ylabel('Velocity (ft/sec)')
         elseif j==3
            title('Response of Pitch Rate (q) to Collective Unit Impulse
Input')
            ylabel('Pitch Rate (rad/sec)')
         elseif j==4
            title('Response of Pitch Angle (theta) to Collective Unit
Impulse Input')
            ylabel('Pitch Angle (rad.)')
         elseif j==5
            title('Response of y-veloctiy (v) to Collective Unit Impulse
Input')
            ylabel('Velocity (ft/sec)')
         elseif j==6
            title('Response of Roll Rate (p) to Collective Unit Impulse
Input')
            ylabel('Roll Rate (rad/sec)')
         elseif j==7
            title('Response of Roll Angle (phi) to Collective Unit
Impulse Input')
            ylabel('Roll Angle (rad)')
         elseif j==8
            title('Response of Yaw Rate (r) to Collective Unit Impulse
Input')
            ylabel('Yaw Rate (rad/sec)')
         elseif j==9
```

```
            title('Response of Yaw Angle (psi) to Collective Unit
Impulse Input')
            ylabel('Yaw Angle(rad)')
         end
      end
   end


elseif u(3)==1
   for j=1:length(C)
      if C(j,j)==1
         figure
         [num,den]=ss2tf(Amat,Bmat(:,1),C(j,:),D);
         [y]=impulse(num,den,t);
         plot(t,y)
         xlabel('time (sec)')
         grid
         if j==1
            title('Response of x-velocity (u) to Lateral Cyclic Unit
Impulse Input')
            ylabel('Velocity (ft/sec)')
         elseif j==2
            title('Response of z-velocity (w) to Lateral Cyclic Unit
Impulse Input')
            ylabel('Velocity (ft/sec)')
         elseif j==3
            title('Response of Pitch Rate (q) to Lateral Cyclic Unit
Impulse Input')
            ylabel('Pitch Rate (rad/sec)')
         elseif j==4
            title('Response of Pitch Angle (theta) to Lateral Cyclic
Unit Impulse Input')
            ylabel('Pitch Angle (rad.)')
         elseif j==5
            title('Response of y-veloctiy (v) to Lateral Cyclic Unit
Impulse Input')
            ylabel('Velocity (ft/sec)')
         elseif j==6
            title('Response of Roll Rate (p) to Lateral Cyclic Unit
Impulse Input')
            ylabel('Roll Rate (rad/sec)')
         elseif j==7
            title('Response of Roll Angle (phi) to Lateral Cyclic Unit
Impulse Input')
            ylabel('Roll Angle (rad)')
         elseif j==8
            title('Response of Yaw Rate (r) to Lateral Cyclic Unit
Impulse Input')
            ylabel('Yaw Rate (rad/sec)')
         elseif j==9
            title('Response of Yaw Angle (psi) to Lateral Cyclic Unit
Impulse Input')
            ylabel('Yaw Angle(rad)')
         end
      end
   end
```

```
elseif u(4)==1

   for j=1:length(C)
      if C(j,j)==1
         figure
         [num,den]=ss2tf(Amat,Bmat(:,1),C(j,:),D);
         [y]=impulse(num,den,t);
         plot(t,y)
         xlabel('time (sec)')
         grid
         if j==1
            title('Response of x-velocity (u) to Pedal Unit Impulse
Input')
            ylabel('Velocity (ft/sec)')
         elseif j==2
            title('Response of z-velocity (w) to Pedal Unit Impulse
Input')
            ylabel('Velocity (ft/sec)')
         elseif j==3
            title('Response of Pitch Rate (q) to Pedal Unit Impulse
Input')
            ylabel('Pitch Rate (rad/sec)')
         elseif j==4
            title('Response of Pitch Angle (theta) to Pedal Unit Impulse
Input')
            ylabel('Pitch Angle (rad.)')
         elseif j==5
            title('Response of y-veloctiy (v) to Pedal Unit Impulse
Input')
            ylabel('Velocity (ft/sec)')
         elseif j==6
            title('Response of Roll Rate (p) to Pedal Unit Impulse
Input')
            ylabel('Roll Rate (rad/sec)')
         elseif j==7
            title('Response of Roll Angle (phi) to Pedal Unit Impulse
Input')
            ylabel('Roll Angle (rad)')
         elseif j==8
            title('Response of Yaw Rate (r) to Pedal Unit Impulse
Input')
            ylabel('Yaw Rate (rad/sec)')
         elseif j==9
            title('Response of Yaw Angle (psi) to Pedal Unit Impulse
Input')
            ylabel('Yaw Angle(rad)')
         end
      end
   end


end
```

15. Save_con.m

```
function save_con()
% This is the machine-generated representation of a MATLAB object
% and its children.  Note that handle values may change when these
% objects are re-created. This may cause problems with some callbacks.
% The command syntax may be supported in the future, but is currently
% incomplete and subject to change.
%
% To re-open this system, just type the name of the m-file at the MATLAB
% prompt. The M-file and its associtated MAT-file must be on your path.

% GUI screen to indicate output files have been saved

% Created for JANRAD version 6.0 by LT David A. Heathorn

load save_con

global H_SAVE_CON

H_SAVE_CON = figure('Color',[0.8 0.8 0.8], ...
      'Colormap',mat0, ...
      'Name','File Save Confirmation', ...
      'NumberTitle','off', ...
      'PointerShapeCData',mat1, ...
      'Position',[119 70 435 193], ...
      'Tag','Fig4');
b = uicontrol('Parent',H_SAVE_CON, ...
      'Units','points', ...
      'BackgroundColor',[0.752941 0.752941 0.752941], ...
      'FontSize',18, ...
      'Position',[65.25 45.75 196.5 60], ...
      'String','Input and Output Files Saved.', ...
      'Style','text', ...
      'Tag','StaticText1');

assignin('base','H_SAVE_CON',H_SAVE_CON);
```

16. Sc_save.m

```
function sc_save()
% This is the machine-generated representation of a MATLAB object
% and its children.  Note that handle values may change when these
% objects are re-created. This may cause problems with some callbacks.
% The command syntax may be supported in the future, but is currently
% incomplete and subject to change.
%
% To re-open this system, just type the name of the m-file at the MATLAB
% prompt. The M-file and its associated MAT-file must be on your path.

% GUI screen to save input and output of Stability and Control Module

% Written for Janrad version 6.0 by LT David A. Heathorn

load sc_save
global H_datain1 H_dataout1 H_SC_SAVE


H_SC_SAVE = figure('Color',[0.8 0.8 0.8], ...
      'Colormap',mat0, ...
      'Name','Save Input and Output', ...
      'NumberTitle','off', ...
      'PointerShapeCData',mat1, ...
      'Position',[120 342 428 198], ...
      'Tag','Fig1');
b = uicontrol('Parent',H_SC_SAVE, ...
      'Units','points', ...
      'BackgroundColor',[0.752941 0.752941 0.752941], ...
      'Position',[25.5 110.25 138 17.25], ...
      'String','Save all input data as ...', ...
      'Style','text', ...
      'Tag','StaticText1');
b = uicontrol('Parent',H_SC_SAVE, ...
      'Units','points', ...
      'BackgroundColor',[0.752941 0.752941 0.752941], ...
      'Position',[25.5 84.75 138 17.25], ...
      'String','Save all output data as ...', ...
      'Style','text', ...
      'Tag','StaticText1');
b = uicontrol('Parent',H_SC_SAVE, ...
      'Units','points', ...
      'BackgroundColor',[0.752941 0.752941 0.752941], ...
      'Position',[254.25 111 45 15], ...
      'String','_in.mat', ...
      'Style','text', ...
      'Tag','StaticText2');
b = uicontrol('Parent',H_SC_SAVE, ...
      'Units','points', ...
      'BackgroundColor',[0.752941 0.752941 0.752941], ...
      'Position',[254.25 87 45 15], ...
      'String','_out.mat', ...
      'Style','text', ...
      'Tag','StaticText2');
H_datain1 = uicontrol('Parent',H_SC_SAVE, ...
      'Units','points', ...
```

```
      'BackgroundColor',[1 1 1], ...
      'Position',[171 109.5 75 16.5], ...
   'Fontsize',12, ...
   'Style','edit', ...
   'String','',...
   'Callback',[...
      'set(gcbo,''String'',get(gcbo,''String''));,'...
      'set(H_dataout1,''String'',get(H_datain1,''String''));,'],...
    'HorizontalAlignment','right',...
    'Tag','EditText1');
H_dataout1 = uicontrol('Parent',H_SC_SAVE, ...
      'Units','points', ...
   'Position',[172.5 87 72 15], ...
   'String','',...
   'Style','text', ...
   'HorizontalAlignment','right',...
      'Tag','StaticText1');
b = uicontrol('Parent',H_SC_SAVE, ...
      'Units','points', ...
      'FontSize',18, ...
      'Position',[25.5 32.25 98.25 29.25], ...
   'String','Save', ...
   'Callback','sc_save_fcn save',...
      'Tag','Pushbutton1');
b = uicontrol('Parent',H_SC_SAVE, ...
      'Units','points', ...
      'FontSize',18, ...
      'Position',[201 30 98.25 29.25], ...
   'String','Continue', ...
   'Callback','sc_save_fcn cont',...
   'Tag','Pushbutton1');

assignin('base','H_datain1',H_datain1);
assignin('base','H_dataout1',H_dataout1);
assignin('base','H_SC_SAVE',H_SC_SAVE);
```

17. sc_save_fcn.m

```
function sc_save_fcn(Action)

%     Switchyard Callback for sc_save.m
%     Written for JANRAD version 6.0 by LT David A. Heathorn

global H_PERF_OUT S_USER_INPUT S_PERF_INPUT S_PERF_OUTPUT S_MATR_VEC...
   H_datain1 H_dataout1 H_SC_SAVE H_SAVE_CON ...
   H_outputfile H_vecfile H_inputfile OUT_COUNT H_SAVE S_STAB_INPUT Amat
Bmat

if nargin
   switch Action
   case 'cont'
      close (H_SC_SAVE)
      close (H_SAVE_CON)
      S_PERF_INPUT=S_USER_INPUT;
      return

   case 'save'

         A=Amat(1:8,1:8);
         B=Bmat(1:8,1:4);
         filename1=get(H_datain1,'String');
         filename1a=[filename1 '_in'];
         eval(['save ',filename1a,' S_USER_INPUT S_STAB_INPUT'])
         unstructure3
         filename2=[filename1 '_out'];
         eval(['save ',filename2,' A B'])
         save_con

   end
end
```

18. sc_status.m

```
function sc_status()
% This is the machine-generated representation of a MATLAB object
% and its children.  Note that handle values may change when these
% objects are re-created. This may cause problems with some callbacks.
% The command syntax may be supported in the future, but is currently
% incomplete and subject to change.
%
% To re-open this system, just type the name of the m-file at the MATLAB
% prompt. The M-file and its associtated MAT-file must be on your path.

% Status screen for Stability and Control Computation

% Written for JANRAD version 6.0 by LT David A. Heathorn

load sc_status
global S_H_SC_STAT

H_SC_STAT = figure('Units','normalized', ...
      'Color',[0.8 0.8 0.8], ...
      'Colormap',mat0, ...
      'Name','Stability & Control Status', ...
      'NumberTitle','off', ...
      'PointerShapeCData',mat1, ...
      'Position',[0.00125 0.045 0.9975 0.89], ...
   'Tag','Fig1');
b = uicontrol('Parent',H_SC_STAT, ...
      'Units','normalized', ...
      'BackgroundColor',[0.752941 0.752941 0.752941], ...
      'Position',[0.10616 0.190385 0.802097 0.767308], ...
      'Style','frame', ...
      'Tag','Frame2');

H_STATUS3 = uicontrol('Parent',H_SC_STAT, ...
      'Units','normalized', ...
      'BackgroundColor',[0.752941 0.752941 0.752941], ...
      'FontSize',12, ...
      'FontWeight','bold', ...
      'Position',[0.27654 0.836538 0.440367 0.0538462], ...
      'String','Stability and Control Analysis Status Box', ...
      'Style','text', ...
      'Tag','StaticText1');
H_STATUS1 = uicontrol('Parent',H_SC_STAT, ...
      'Units','normalized', ...
      'BackgroundColor',[0.752941 0.752941 0.752941], ...
      'FontSize',12, ...
      'FontWeight','bold', ...
      'Position',[0.279161 0.678846 0.439056 0.134615], ...
      'Style','text', ...
      'Tag','StaticText2');
H_STATUS2 = uicontrol('Parent',H_SC_STAT, ...
      'Units','normalized', ...
      'BackgroundColor',[0.752941 0.752941 0.752941], ...
      'FontSize',12, ...
      'FontWeight','bold', ...
      'Position',[0.281782 0.526923 0.436435 0.140385], ...
```

```
      'Style','text', ...
      'Tag','StaticText3');
H_STATUS = uicontrol('Parent',H_SC_STAT, ...
      'Units','normalized', ...
      'BackgroundColor',[0.752941 0.752941 0.752941], ...
      'FontSize',12, ...
      'FontWeight','bold', ...
      'Position',[0.284404 0.369231 0.433814 0.15], ...
      'Style','text', ...
      'Tag','StaticText4');
H_STATUS4 = uicontrol('Parent',H_SC_STAT, ...
      'Units','normalized', ...
      'BackgroundColor',[0.752941 0.752941 0.752941], ...
      'FontSize',12, ...
      'FontWeight','bold', ...
      'Position',[0.281782 0.203846 0.433814 0.151923], ...
      'Style','text', ...
      'Tag','StaticText5');
H_BK = uicontrol('Parent',H_SC_STAT, ...
      'Units','normalized', ...
      'Callback','sc_anal_fcn back', ...
      'FontSize',12, ...
      'FontWeight','bold', ...
      'Position',[0.0767857 0.064 0.178571 0.072], ...
      'String','<< Back', ...
      'Tag','Pushbutton1');
H_GO = uicontrol('Parent',H_SC_STAT, ...
      'Units','normalized', ...
      'Callback','global S_H_SC_STAT; sc_status_fcn anal', ...
      'FontSize',12, ...
      'FontWeight','bold', ...
      'Position',[0.301786 0.0613333 0.178571 0.072], ...
      'String','Analyze', ...
      'Tag','Pushbutton2');
H_RUPT = uicontrol('Parent',H_SC_STAT, ...
      'Units','normalized', ...
      'Callback','sc_anal_fcn interrupt', ...
      'Enable','off', ...
      'FontSize',12, ...
      'FontWeight','bold', ...
      'Position',[0.528571 0.0613333 0.178571 0.072], ...
      'String','Interrupt', ...
      'Tag','Pushbutton3');
H_RES = uicontrol('Parent',H_SC_STAT, ...
      'Units','normalized', ...
      'Callback','sc_anal_fcn resume', ...
      'Enable','off', ...
      'FontSize',12, ...
      'FontWeight','bold', ...
      'Position',[0.755357 0.0613333 0.178571 0.072], ...
      'String','Resume', ...
      'Tag','Pushbutton4');
H_MEN = uimenu('Parent',H_SC_STAT, ...
      'Label','JANRAD Options', ...
      'Tag','uimenu1');
c = uimenu('Parent',H_MEN, ...
      'Callback','sc_anal_fcn quit', ...
```

```
      'Label','Quit JANRAD', ...
      'Tag','JANRAD OptionsSubuimenu1');
c = uimenu('Parent',H_MEN, ...
      'Callback','sc_anal_fcn return', ...
      'Label','Return to Begining', ...
      'Tag','JANRAD OptionsSubuimenu1');
c = uimenu('Parent',H_MEN, ...
      'Callback','sc_status_fcn delta_input', ...
      'Label','Change Input Parameters', ...
      'Tag','Subuimenu1');
c = uimenu('Parent',H_MEN, ...
      'Callback','sc_status_fcn about', ...
      'Label','About Janrad 98 ...', ...
      'Separator','on', ...
   'Tag','Subuimenu1');

S_H_SC_STAT.h_sc_stat=H_SC_STAT;
S_H_SC_STAT.h_status=H_STATUS;
S_H_SC_STAT.h_status1=H_STATUS1;
S_H_SC_STAT.h_status2=H_STATUS2;
S_H_SC_STAT.h_status3=H_STATUS3;
S_H_SC_STAT.h_status4=H_STATUS4;
S_H_SC_STAT.h_bk=H_BK;
S_H_SC_STAT.h_go=H_GO;
S_H_SC_STAT.h_rupt=H_RUPT;
S_H_SC_STAT.h_res=H_RES;
S_H_SC_STAT.h_men=H_MEN;
```

19. sc_status_fcn.m

```
function sc_status_fcn(Action)

% Switchyard Callback for sc_status.m
% Written for JANRAD version 6.0 by LT David A. Heathorn

global  S_H_SC_STAT S_PERF_INPUT S_STAB_INPUT_1 S_STAB_INPUT_2

% load input_param
H_SC_STAT=S_H_SC_STAT.h_sc_stat;
H_STATUS=S_H_SC_STAT.h_status;
H_STATUS=S_H_SC_STAT.h_status1;
H_STATUS=S_H_SC_STAT.h_status2;
H_STATUS=S_H_SC_STAT.h_status3;
H_STATUS=S_H_SC_STAT.h_status4;
H_BK=S_H_SC_STAT.h_bk;
H_GO=S_H_SC_STAT.h_go;
H_RUPT=S_H_SC_STAT.h_rupt;
H_RES=S_H_SC_STAT.h_res;
H_MEN=S_H_SC_STAT.h_men;


if nargin,
   switch Action

   case 'back'
      stability_control_input_2
      close(H_SC_STAT)
   case 'anal'
      set(H_GO,'Enable','off');
      set(H_RUPT,'Enable','on');
      set(H_BK,'Enable','off');
      set(H_RES,'Enable','off');
      set(H_MEN,'Enable','off');

      if S_PERF_INPUT.Vinf<20
          Hover
       elseif S_PERF_INPUT.Vinf >=20
          Cruise
      end

      close(H_SC_STAT)

   case 'interrupt'
      set(H_GO,'Enable','off');
      set(H_RUPT,'Enable','off');
      set(H_BK,'Enable','off');
      set(H_RES,'Enable','on');
      set(H_MEN,'Enable','on');
      uiwait;
   case 'resume'
      set(H_GO,'Enable','off');
      set(H_RUPT,'Enable','on');
      set(H_BK,'Enable','cff');
      set(H_RES,'Enable','off');
      set(H_MEN,'Enable','off');
```

```
      uiresume;
   case 'quit'
      quit_gui
   case 'return'
      close (H_SC_STAT)
      clear all
      janrad98
   case 'delta_input'
      close (H_SC_STAT)
      stabilty_control_input_1
   case 'about'
      about_janrad
   end
end
```

20. Stab_calc_la.m

```
%  Stab_calc_la.m

%     Calculates change in CT CH CQ a1s b1s with respect to change in
inflow ratio.

%     Written for JANRAD version 6.0 - LT David A. Heathorn

global RADSPC_VAL NL_TWIST_VAL NEW_AUX_VAL FIX_TPP_VAL NEW_TPP
S_PERF_OUTPUT lamdaT_trim


set(H_STATUS,'String','EXECUTING ROTOR DERIVATIVE ROUTINE')
set(H_STATUS1,'String',['RUN ELAPSED TIME IS ' num2str(fix(toc)) '
SECONDS'])

pause(3)

lamdasave=lamdaT_trim

for kk=1:3

   if kk==1

      lamda_prime=lamdasave-(.5*lamdasave);
      n=1;
      vlamp_la(1,n)=-lamda_prime;

   elseif kk==2

      n=2;
      lamda_prime=lamdasave;
      vlamp_la(1,n)=-lamda_prime;

   else

      lamda_prime=lamdasave+(.5*lamdasave);
      n=3;
      vlamp_la(1,n)=-lamda_prime;

   end

   unstructure2
   theta1c=theta1c/57.3;
   theta1s=theta1s/57.3;
   thetao=thetao/57.3;
   alphaT=alphaT/57.3;
   betao=betao/57.3;

   rho=.002377*(-.000031*PA+(-.002*temp+1.118));

   q=0.5*rho*Vinf^2;

   Adisk=pi*R^2;
```

```
   Vtip=omega*R;
   temp_rank=temp+459.67;
   spd_snd=49.1*sqrt(temp_rank);
   Lwing=q*CLwing*Swing;

   Dfuse=q*Afh;

   CDwing=CDowing+(CLwing^2/(ewing*pi*(bwing^2/Swing)));

   CDhoriz=CDohoriz+(CLhoriz^2/(.8*pi*(bhoriz^2/Shoriz)));

   CDvert=CDovert+(CLvert^2/(.8*pi*(bvert^2/Svert)));

   Dwing=q*CDwing*Swing;

   Dhoriz=q*CDhoriz*Shoriz;

   Dvert=q*CDvert*Svert;
   Dftotal=(Dfuse+Dwing+Dhoriz+Dvert);

   % This section is for compound helos, adjustment of aux thrust
efficiency with airspeed
   if Vinf/1.68781<=70
      AUXEFF=.650;
   elseif Vinf/1.68781<=100
      AUXEFF=.65+.0025*((Vinf/1.68781)-70);
   elseif Vinf/1.68781<=160
      AUXEFF=.725+.0025*((Vinf/1.68781)-100);
   elseif Vinf/1.68781<=210
      AUXEFF=.85+.00007*((Vinf/1.68781)-160);
   else
      AUXEFF=.847;
   end

   % This section provided aux thrust schedule for compound helo
   switch NEW_AUX_VAL
   case 0
      Dftotal=(Dfuse+Dwing+Dhoriz+Dvert)-Taux;
   case 1
      if PA==8000
         Taux=16*Vinf/1.68781; % linear increase in aux thrust up to 210
knots
         S_PERF_INPUT.Taux=Taux;
         S_USER_INPUT.Taux=Taux;
      elseif PA==5000
         Taux=17.024*Vinf/1.68781;  % linear increase in aux thrust up
to 210 knots
         S_PERF_INPUT.Taux=Taux;
         S_USER_INPUT.Taux=Taux;
      elseif PA==0
         Taux=Dftotal;
         S_PERF_INPUT.Taux=Taux;
         S_USER_INPUT.Taux=Taux;
      end
   end
   Lhoriz=q*CLhoriz*Shoriz;
```

```
Lvert=q*CLvert*Svert;

Lftotal=Lwing+Lhoriz;
if FIX_TPP_VAL==1
   alphaT=NEW_TPP;      %set tip path angle
else

   Drotor=Hrotor;        % keep the tip path plane angle constant
   alphaT_trim=alphaT;
   altpp=alphaT;

end

%   *** thrust calculation ***


T=(GW-Lftotal)/cos(alphaT);


CT=T/(Adisk*rho*Vtip^2);


%   *** setup blade radius elements, azimuth elements,

%       induced velocity distributions, and determination

%       of coning angle and tip loss parameter ***


B=1-(sqrt(2*CT)/b);

Reff=B*R;

Rbar=Reff-e;


if RADSPC_VAL==1
   NEW_r1=[NEW_r, Reff/R];
   n=length(NEW_r1);
   dr=diff(NEW_r1)*R;
   r=(NEW_r1(1:n-1)*R)+dr/2;
else
   dr=(Reff-grip)/nbe;
   r=grip:dr:Reff-dr;,r=r+dr/2;
end
if NL_TWIST_VAL==1
   NL_TWIST=NL_TWIST/57.3;
   n=length(NL_TWIST);
   if RADSPC_VAL==1
      y=((Reff/R)-NEW_r(n))*((NL_TWIST(n)-NL_TWIST(n-1))/(NEW_r(n)-
NEW_r(n-1)));
   else
      y=((Reff/R)-r(n))*((NL_TWIST(n)-NL_TWIST(n-1))/(r(n)- r(n-1)));
   end
   NL_TWIST1=[NL_TWIST (NL_TWIST(n)+y)];
   m=length(NL_TWIST1);
   dTW=diff(NL_TWIST1);
```

```
      twist=(NL_TWIST1(1:m-1))+dTW/2;
      betat=twist;
   else
      betat=twist*(0.7-(r/R));

   end
   rT1=rT2;              % *** Set value for  rT as calculated for tirm
***

   RbarT=rT1*Rbar;

   mblade=wblade/32.17;

   %betao=asin((T/b*RbarT-(.5*(R-e)+e)*wblade)/((.5*(R-
e)+e)^2*omega^2*mblade))

   psi=0:360/naz:360-360/naz;,psi=psi'/57.3;



   %% set up vector of blade element chords and then varies them as

   %%  requested with the blade taper and blade taper start position

   %%  rchord=root chord

   %%  cblade=vector of blade element chord lengths

   %%  tr=taper ratio (tip/root)

   %%  trst=taper ratio start position (r/R)


   cblade=rchord*ones(size(r));     % gives all elements same chord
length initially


   if tr==0 % prevents division by zero later in code

      tr=1; %  in case 0 is enter for taper ratio instead

   end            %  of 1 for no taper


   if trst==0

      slope=(rchord-rchord*tr)/(Reff-grip);     % Modifies each element

      cblade=cblade-slope*(r-grip);       %  chord length wrt input

      tchord=cblade(nbe);                         %  taper ratio which has
been
```

```
      mchord=sum(cblade)/nbe;              %  been converted into a slope

      %  top portion takes into

   else                                    %  account the possibility
that

      slope=(rchord-rchord*tr)/(R*(1-trst));    %  a 0 start position is
really at

      z=fix(nbe*trst);                     %  the start of the aero
portion

      if z<=1           % prevents beginning index fm being zero

         z=1;

      end

      cblade(z:nbe)=cblade(z:nbe)-(r(z:nbe)-r(z))*slope;

      tchord=cblade(nbe);

      mchord=sum(cblade)/nbe;

   end



   %   *** induced velocity determination ***



   vi_trim=lamdaT_trim*Vtip+(Vinf*sin(alphaT_trim));
   vi=lamda_prime*Vtip+(Vinf*sin(alphaT));

   % Calulate new coning angle based on change in alpha of blade

   phi_old=atan((Vinf*sin(alphaT_trim)+vi_trim)/Vtip);
   phi_new=atan((Vinf*sin(alphaT)+vi)/Vtip);

   betao=betao*(1-((phi_new-phi_old)/(thetao-phi_old)));

   vi=vi*ones(size(r));


   %   *** Calculate theta based on trim conditions ***


   theta=thetao+theta1c.*cos(psi)+theta1s.*sin(psi);
```

```
   %   *** rotor trimming routine ***


   set(H_STATUS,'String','CALULATING ')
   set(H_STATUS1,'String',['RUN ELAPSED TIME IS ' num2str(fix(toc)) '
SECONDS'])
   pause(3)
   set(H_STATUS2,'String','')


   Tpsi=zeros(size(psi));
   Npsi=zeros(size(psi));
   thrcalc

   Trotor=mean(Tpsi)*b;

   Mpsi(:,k)=zeros(size(psi));

   tmcalc




   % *** calculating drag moments ***
   set(H_STATUS2,'String','CALCULATING DRAG MOMENT')
   set(H_STATUS1,'String',['RUN ELAPSED TIME IS ' num2str(fix(toc)) '
SECONDS'])
   pause(2)
   DMpsi=zeros(size(psi));

   dmcalc
   Qrotor=mean(DMpsi)*b;

   %   *** calculating rotor H force ***

   set(H_STATUS2,'String','CALCULATING ROTOR DRAG')
   set(H_STATUS1,'String',['RUN ELAPSED TIME IS ' num2str(fix(toc)) '
SECONDS'])
   pause(2)


   dT=[dT ddT];
   dN=[dN ddN];
   dD=[dD ddD];

   for i=1:length(r)+1,

      H1c(i)=2*sum(dT(:,i).*cos(psi))/naz;

      H1s(i)=2*sum(dD(:,i).*sin(psi))/naz;

   end
```

```
Hrotor=(((b*cos(alphaT)/2)*(sum(H1s)-sin(betao)*sum(H1c)))+Drotor)/2;



CT=Trotor/(Adisk*rho*Vtip^2);
CH=Hrotor/(Adisk*rho*Vtip^2);
CQ=Qrotor/(Adisk*rho*Vtip^2*R);

solidity=b*(sum(cblade)/length(r))/(pi*R);

vctsig_la(1,n)=CT/solidity;
vcqsig_la(1,n)=CQ/solidity;
vchsig_la(1,n)=CH/solidity;
vmu_la(1,n)=mu;



% Determine flapping

lockno=rho*a*(sum(cblade)/length(r))*R^4/Ib;

A1=-theta1c;
B1=-theta1s;
ctsig=vctsig_la(1,n);
thetao=thetao-.7*theta1;


altpp=(vlamp_la(1,n)/mu)+(solidity*ctsig/2/(mu*sqrt(mu^2+vlamp_la(1,n)^2
)));

num1=thetao*mu*8/3;
num2=2*theta1*mu;
num3=-B1*(1+((3/2)*mu^2));
num4=2*mu*(mu*altpp-(ctsig*solidity/2/mu));
den1=1-((mu^2)/2);
term1=(num1+num2+num3)/den1;

num5=12*(e/R);
den5=lockno*((1-(e/R))^3)*(1+(mu^4)/4);
term2=num5/den5;

num6=A1*(1+(.5*mu^2));
num7=(4/3)*ctsig;
num8=(2/3)*mu*lockno/a;
num9=solidity/2*mu;
den6=1+(3*e/2/R);
term3=num6+(num7*((num8/den6)+num9));

va1s_la(1,n)=(term1+(term2*term3));


vvi_la(1,n)=mean(vi);
```

```
   bnum1=(4/3)*ctsig;
   bden1=1+((mu^2)/2);
   bnum2=(2/3)*mu*lockno/a;
   bden2=1+(3*e/2/R);
   bterm1=solidity/2/mu;
   bterm2=A1+((bnum1/bden1)*((bnum2/bden2)+bterm1));
   bnum3=12*e/R;
   bden3=lockno*((1-(e/R))^3)*(1-(.25*(mu^4)));
   bterm3=bnum3/bden3;
   bnum4=2*mu*((mu*(altpp-altpp))-(ctsig*solidity/2/mu));

   bterm4=num1+num2+num3+bnum4;

   vb1s_la(1,n)=bterm2+(bterm3*bterm4);



   %tail rotor
   %
   if tailrot==1
      ctsig=vctsig_la(1,n);
      cqsig=vcqsig_la(1,n);
      ohm=omega;
      lv=lvd-xcg;
      g=32.2;
      A=pi*R*R;


      Abt=bt*Rt*ct;
      At=pi*Rt*Rt;
      mut=Vinf/ohmt/Rt;
      sigmat=Abt/At;
      lt=ltd-xcg;
      ht=htd-zcg;
      yt=ytd-ycg;
      locknot=rho*at*ct*Rt^4/Ibt;
      Tt=(cqsig*solidity*rho*A*(ohm*R)^2*R-Lvert*lv)/lt;
      ctsigt=Tt/(rho*At*(ohmt*Rt)^2);
      lampt=-ctsigt*solidity/2/mu;
      aot=2/3*locknot*ctsigt/at-3/2*g*Rt^2/(ohmt*Rt)^2;
      a1st=(-2*((4/3*mut*aot+ctsigt*sigmat/2/mut)/(2+mut*mut))*(1-
(4*mut/(2+3*mut*mut))^2)*...
         tan(delta3))/(((2-mut*mut)/(2+3*mut*mut))+(1-
(4*mut/(2+3*mut*mut))^2)*tan(delta3)^2);

b1st=2*((4/3*mu*aot*ctsigt*solidity/2/mu)/(2+mu*mu))+a1st*tan(delta3);
      thetaot=((4*ctsigt/at)-
(.5+mut*mut/2)*theta1t+mut*b1st*tan(delta3)-lampt)/...
         (2/3+mut*mut)-aot*tan(delta3);
      theta75t=thetaot+.75*theta1t;
      v1t=sqrt(Tt/2/rho/At);
      vctsigt_lat(1,n)=ctsigt;
      vlampt(1,n)=lampt;

   end
```

```
end
```

21. Stab_calc_mu.m

```
% Stab_calc_mu.m

%    Calculates change in CT CH CQ a1s b1s with respect to input
parameters

%    Created for JANRAD version 6.0 by LT David A. Heathorn

global RADSPC_VAL NL_TWIST_VAL NEW_AUX_VAL FIX_TPP_VAL NEW_TPP
S_PERF_OUTPUT lamdaT_trim

set(H_STATUS,'String','EXECUTING ROTOR DERIVATIVE ROUTINE')
set(H_STATUS1,'String',['RUN ELAPSED TIME IS ' num2str(fix(toc)) '
SECONDS'])

pause(3)

%  *** calculation of required parameters ***


Vsave=Vinf;

for kk=1:3

   if kk==1
      Vinf=Vsave-(Vsave*.15);
      n=1;

   elseif kk==2
      Vinf=Vsave;
      n=2;

   else
      Vinf=Vsave+(Vsave*.15);
      n=3;
   end

   unstructure2
   theta1c=theta1c/57.3;
   theta1s=theta1s/57.3;
   thetao=thetao/57.3;
   alphaT=alphaT/57.3;
   betao=betao/57.3;


   rho=.002377*(-.000031*PA+(-.002*temp+1.118));

   q=0.5*rho*Vinf^2;

   Adisk=pi*R^2;

   Vtip=omega*R;
   temp_rank=temp+459.67;
   spd_snd=49.1*sqrt(temp_rank);
   Lwing=q*CLwing*Swing;
```

```
   Dfuse=q*Afh;

   CDwing=CDowing+(CLwing^2/(ewing*pi*(bwing^2/Swing)));

   CDhoriz=CDohoriz+(CLhoriz^2/(.8*pi*(bhoriz^2/Shoriz)));

   CDvert=CDovert+(CLvert^2/(.8*pi*(bvert^2/Svert)));

   Dwing=q*CDwing*Swing;

   Dhoriz=q*CDhoriz*Shoriz;

   Dvert=q*CDvert*Svert;
   Dftotal=(Dfuse+Dwing+Dhoriz+Dvert);

   % This section is for compound helos, adjustment of aux thrust
efficiency with airspeed
   if Vinf/1.68781<=70
      AUXEFF=.650;
   elseif Vinf/1.68781<=100
      AUXEFF=.65+.0025*((Vinf/1.68781)-70);
   elseif Vinf/1.68781<=160
      AUXEFF=.725+.0025*((Vinf/1.68781)-100);
   elseif Vinf/1.68781<=210
      AUXEFF=.85+.00007*((Vinf/1.68781)-160);
   else
      AUXEFF=.847;
   end

   % This section provided aux thrust schedule for compound helo
   switch NEW_AUX_VAL
   case 0
      Dftotal=(Dfuse+Dwing+Dhoriz+Dvert)-Taux;
   case 1
      if PA==8000
         Taux=16*Vinf/1.68781;  % linear increase in aux thrust up to
210 knots
         S_PERF_INPUT.Taux=Taux;
         S_USER_INPUT.Taux=Taux;
      elseif PA==5000
         Taux=17.024*Vinf/1.68781;  % linear increase in aux thrust up
to 210 knots
         S_PERF_INPUT.Taux=Taux;
         S_USER_INPUT.Taux=Taux;
      elseif PA==0
         Taux=Dftotal;
         S_PERF_INPUT.Taux=Taux;
         S_USER_INPUT.Taux=Taux;
      end
   end
   Lhoriz=q*CLhoriz*Shoriz;

   Lvert=q*CLvert*Svert;

   Lftotal=Lwing+Lhoriz;
   if FIX_TPP_VAL==1
      alphaT=NEW_TPP;      %set tip path angle
```

```
else
   Drotor=Hrotor;
   alphaT=atan2((Dftotal+Drotor),(GW-Lftotal));
end

mu=Vinf*cos(alphaT)/Vtip;

%   *** thrust calculation ***


T=(GW-Lftotal)/cos(alphaT);


CT=T/(Adisk*rho*Vtip^2);


%   *** setup blade radius elements, azimuth elements,

%       induced velocity distributions, and determination

%       of coning angle and tip loss parameter ***


B=1-(sqrt(2*CT)/b);

Reff=B*R;

Rbar=Reff-e;


if RADSPC_VAL==1
   NEW_r1=[NEW_r, Reff/R];
   n=length(NEW_r1);
   dr=diff(NEW_r1)*R;
   r=(NEW_r1(1:n-1)*R)+dr/2;
else
   dr=(Reff-grip)/nbe;
   r=grip:dr:Reff-dr;,r=r+dr/2;
end
if NL_TWIST_VAL==1
   NL_TWIST=NL_TWIST/57.3;
   n=length(NL_TWIST);
   if RADSPC_VAL==1
      y=((Reff/R)-NEW_r(n))*((NL_TWIST(n)-NL_TWIST(n-1))/(NEW_r(n)-
NEW_r(n-1)));
   else
      y=((Reff/R)-r(n))*((NL_TWIST(n)-NL_TWIST(n-1))/(r(n)- r(n-1)));
   end
   NL_TWIST1=[NL_TWIST (NL_TWIST(n)+y)];
   m=length(NL_TWIST1);
   dTW=diff(NL_TWIST1);
   twist=(NL_TWIST1(1:m-1))+dTW/2;
   betat=twist;
else
   betat=twist*(0.7-(r/R));

end
```

```
rT1=rT2;,% *** Set value for  rT as calculated for tirm ***

RbarT=rT1*Rbar;

mblade=wblade/32.17;

% Coning angle using Thrust given from Trim condition


betao=asin((T/b*RbarT-(.5*(R-e)+e)*wblade)/((.5*(R-
e)+e)^2*omega^2*mblade));

psi=0:360/naz:360-360/naz;,psi=psi'/57.3;


%% set up vector of blade element chords and then varies them as

%%  requested with the blade taper and blade taper start position

%%  rchord=root chord

%%  cblade=vector of blade element chord lengths

%%  tr=taper ratio (tip/root)

%%  trst=taper ratio start position (r/R)


cblade=rchord*ones(size(r));     % gives all elements same chord
length initially


if tr==0 % prevents division by zero later in code

   tr=1; %  in case 0 is enter for taper ratio instead

end            %  of 1 for no taper


if trst==0

   slope=(rchord-rchord*tr)/(Reff-grip);     % Modifies each element

   cblade=cblade-slope*(r-grip);       %  chord length wrt input

   tchord=cblade(nbe);                           %  taper ratio which has
been

   mchord=sum(cblade)/nbe;             %  been converted into a slope

   %  top portion takes into
```

```
   else                                          %  account the possibility
that

      slope=(rchord-rchord*tr)/(R*(1-trst));    %  a 0 start position is
really at

      z=fix(nbe*trst);                           %  the start of the aero
portion

      if z<=1            % prevents beginning index fm being zero

         z=1;

      end

      cblade(z:nbe)=cblade(z:nbe)-(r(z:nbe)-r(z))*slope;

      tchord=cblade(nbe);

      mchord=sum(cblade)/nbe;

   end



   %   *** induced velocity determination ***

   % Wheatley Eqn for Fwd flt

   lamdaT=0;

   lamda=1;

   while abs(lamdaT-lamda)>1e-4

      lamda=lamdaT;

      lamdaT=mu*sin(alphaT)+0.5*CT/sqrt(lamdaT^2+mu^2);

   end

   vi=lamdaT*Vtip-Vinf*sin(alphaT);

   vi=vi*ones(size(r));

   %   *** Calculate theta based on trim conditions ***
```

```
   theta=thetao+theta1c.*cos(psi)+theta1s.*sin(psi);



   %   *** rotor trimming routine ***



   set(H_STATUS,'String','CALULATING ')
   set(H_STATUS1,'String',['RUN ELAPSED TIME IS ' num2str(fix(toc)) '
SECONDS'])
   pause(3)
   set(H_STATUS2,'String','')


   Tpsi=zeros(size(psi));
   Npsi=zeros(size(psi));
   thrcalc

   Trotor=mean(Tpsi)*b;

   Mpsi(:,k)=zeros(size(psi));

   tmcalc


   % *** calculating drag moments ***

   set(H_STATUS2,'String','CALCULATING DRAG MOMENT')
   set(H_STATUS1,'String',['RUN ELAPSED TIME IS ' num2str(fix(toc)) '
SECONDS'])
   pause(2)
   DMpsi=zeros(size(psi));

   dmcalc
   Qrotor=mean(DMpsi)*b;

   %   *** calculating rotor H force ***

   set(H_STATUS2,'String','CALCULATING ROTOR DRAG')
   set(H_STATUS1,'String',['RUN ELAPSED TIME IS ' num2str(fix(toc)) '
SECONDS'])
   pause(2)


   dT=[dT ddT];
   dN=[dN ddN];
   dD=[dD ddD];

   for i=1:length(r)+1,

      H1c(i)=2*sum(dT(:,i).*cos(psi))/naz;

      H1s(i)=2*sum(dD(:,i).*sin(psi))/naz;

   end
```

```
Hrotor=(((b*cos(alphaT)/2)*(sum(H1s)-sin(betao)*sum(H1c)))+Drotor)/2;


CT=Trotor/(Adisk*rho*Vtip^2);
CH=Hrotor/(Adisk*rho*Vtip^2);
CQ=Qrotor/(Adisk*rho*Vtip^2*R);

solidity=b*(sum(cblade)/length(r))/(pi*R);

vctsig_mu(1,n)=CT/solidity;
vcqsig_mu(1,n)=CQ/solidity;
vchsig_mu(1,n)=CH/solidity;
vmu_mu(1,n)=mu;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%NOT SURE AFTER
THIS5%%%%%%%%%%%%%%%%%%%%%%%%%%%%%


altpp=-alphaT;
thetao=thetao-.7*theta1;
lockno=rho*a*(sum(cblade)/length(r))*R^4/Ib;

A1=-theta1c;
B1=-theta1s;
ctsig=vctsig_mu(1,n);

num1=thetao*mu*8/3;
num2=2*theta1*mu;
num3=-B1*(1+((3/2)*mu^2));
num4=2*mu*(mu*altpp-(ctsig*solidity/2/mu));
den1=1-((mu^2)/2);
term1=(num1+num2+num3)/den1;

num5=12*(e/R);
den5=lockno*((1-(e/R))^3)*(1+(mu^4)/4);
term2=num5/den5;

num6=A1*(1+(.5*mu^2));
num7=(4/3)*ctsig;
num8=(2/3)*mu*lockno/a;
num9=solidity/2*mu;
den6=1+(3*e/2/R);
term3=num6+(num7*((num8/den6)+num9));

va1s_mu(1,n)=(term1+(term2*term3));

vvi_mu(1,n)=mean(vi);

bnum1=(4/3)*ctsig;
bden1=1+((mu^2)/2);
bnum2=(2/3)*mu*lockno/a;
bden2=1+(3*e/2/R);
bterm1=solidity/2/mu;
bterm2=A1+((bnum1/bden1)*((bnum2/bden2)+bterm1));
bnum3=12*e/R;
bden3=lockno*((1-(e/R))^3)*(1-(.25*(mu^4)));
```

```
   bterm3=bnum3/bden3;
   bnum4=2*mu*((mu*(altpp-altpp))-(ctsig*solidity/2/mu));

   bterm4=num1+num2+num3+bnum4;

   vb1s_mu(1,n)=bterm2+(bterm3*bterm4);


   %tail rotor   %
   if tailrot==1
      ctsig=vctsig_mu(1,n);
      cqsig=vcqsig_mu(1,n);
      ohm=omega;
      lv=lvd-xcg;
      g=32.2;
      A=pi*R*R;


      Abt=bt*Rt*ct;
      At=pi*Rt*Rt;
      mut=Vinf/ohmt/Rt;
      sigmat=Abt/At;
      lt=ltd-xcg;
      ht=htd-zcg;
      yt=ytd-ycg;
      locknot=rho*at*ct*Rt^4/Ibt;
      Tt=(cqsig*solidity*rho*A*(ohm*R)^2*R-Lvert*lv)/lt;
      ctsigt=Tt/(rho*At*(ohmt*Rt)^2);
      lampt=-ctsigt*solidity/2/mu;
      aot=2/3*locknot*ctsigt/at-3/2*g*Rt^2/(ohmt*Rt)^2;
      a1st=(-2*((4/3*mut*aot+ctsigt*sigmat/2/mut)/(2+mut*mut))*(1-
(4*mut/(2+3*mut*mut))^2)*...
         tan(delta3))/(((2-mut*mut)/(2+3*mut*mut))+(1-
(4*mut/(2+3*mut*mut))^2)*tan(delta3)^2);

b1st=2*((4/3*mu*aot*ctsigt*solidity/2/mu)/(2+mu*mu))+a1st*tan(delta3);
      thetaot=((4*ctsigt/at)-
(.5+mut*mut/2)*theta1t+mut*b1st*tan(delta3)-lampt)/...
         (2/3+mut*mut)-aot*tan(delta3);
      theta75t=thetaot+.75*theta1t;
      v1t=sqrt(Tt/2/rho/At);
      vctsigt_mut(1,n)=ctsigt;
      vmut(1,n)=mut;


   end



end
```

22. Stab_cal_to.m

```
% Stab_calc_to.m

%     Calculates change in CT CH CQ a1s b1s with respect to input
parameters

%     Written for JANRAD version 6.0 by LT David A. Heathorn

global RADSPC_VAL NL_TWIST_VAL NEW_AUX_VAL FIX_TPP_VAL NEW_TPP
S_PERF_OUTPUT lamdaT_trim


set(H_STATUS,'String','EXECUTING ROTOR DERIVATIVE ROUTINE')
set(H_STATUS1,'String',['RUN ELAPSED TIME IS ' num2str(fix(toc)) '
SECONDS'])

pause(3)
%   *** calculation of required parameters ***



for kk=1:3

   if kk==1
      unstructure2
      thetao=thetao/57.3;
      theta_save=thetao;
      theta_inc=-(1/57.3);
      thetao=thetao+(theta_inc);
      n=1;
      vthetao_to(1,n)=thetao;

   elseif kk==2
      unstructure2
      thetao=thetao/57.3;
      theta_inc=0;
      n=2;
      vthetao_to(1,n)=thetao;

   else
      unstructure2
      thetao=thetao/57.3;
      theta_inc=(1/57.3);
      thetao=thetao+(theta_inc);
      n=3;
      vthetao_to(1,n)=thetao;

   end

   theta1c=theta1c/57.3;
   theta1s=theta1s/57.3;
   alphaT=alphaT/57.3;
   betao=betao/57.3;

   Vinf=Vsave;
```

```
   rho=.002377*(-.000031*PA+(-.002*temp+1.118));

   q=0.5*rho*Vinf^2;

   Adisk=pi*R^2;

   Vtip=omega*R;
   temp_rank=temp+459.67;
   spd_snd=49.1*sqrt(temp_rank);
   Lwing=q*CLwing*Swing;

   Dfuse=q*Afh;

   CDwing=CDowing+(CLwing^2/(ewing*pi*(bwing^2/Swing)));

   CDhoriz=CDohoriz+(CLhoriz^2/(.8*pi*(bhoriz^2/Shoriz)));

   CDvert=CDovert+(CLvert^2/(.8*pi*(bvert^2/Svert)));

   Dwing=q*CDwing*Swing;

   Dhoriz=q*CDhoriz*Shoriz;

   Dvert=q*CDvert*Svert;
   Dftotal=(Dfuse+Dwing+Dhoriz+Dvert);

   % This section is for compound helos, adjustment of aux thrust
efficiency with airspeed

   if Vinf/1.68781<=70
      AUXEFF=.650;
   elseif Vinf/1.68781<=100
      AUXEFF=.65+.0025*((Vinf/1.68781)-70);
   elseif Vinf/1.68781<=160
      AUXEFF=.725+.0025*((Vinf/1.68781)-100);
   elseif Vinf/1.68781<=210
      AUXEFF=.85+.00007*((Vinf/1.68781)-160);
   else
      AUXEFF=.847;
   end

   % This section provided aux thrust schedule for compound helo
   switch NEW_AUX_VAL
   case 0
      Dftotal=(Dfuse+Dwing+Dhoriz+Dvert)-Taux;
   case 1
      if PA==8000
         Taux=16*Vinf/1.68781  % linear increase in aux thrust up to 210
knots
         S_PERF_INPUT.Taux=Taux;
         S_USER_INPUT.Taux=Taux;
      elseif PA==5000
         Taux=17.024*Vinf/1.68781  % linear increase in aux thrust up to
210 knots
         S_PERF_INPUT.Taux=Taux;
         S_USER_INPUT.Taux=Taux;
      elseif PA==0
```

```
      Taux=Dftotal;
      S_PERF_INPUT.Taux=Taux;
      S_USER_INPUT.Taux=Taux;
   end
end
Lhoriz=q*CLhoriz*Shoriz;

Lvert=q*CLvert*Svert;

Lftotal=Lwing+Lhoriz;
if FIX_TPP_VAL==1
   Drotor=Hrotor;
   alphaT=NEW_TPP;      %set tip path angle
else
   lamdaT=lamdaT_trim;


   altpp=asin((lamdaT-(.5*CT/sqrt(lamdaT^2+mu^2)))/mu);
   alphaT=altpp;

   Drotor=Hrotor;

end


%   *** thrust calculation ***


T=(GW-Lftotal)/cos(alphaT);


CT=T/(Adisk*rho*Vtip^2);


%   *** setup blade radius elements, azimuth elements,

%       induced velocity distributions, and determination

%       of coning angle and tip loss parameter ***


B=1-(sqrt(2*CT)/b);

Reff=B*R;

Rbar=Reff-e;


if RADSPC_VAL==1
   NEW_r1=[NEW_r, Reff/R];
   n=length(NEW_r1);
   dr=diff(NEW_r1)*R;
   r=(NEW_r1(1:n-1)*R)+dr/2;
else
   dr=(Reff-grip)/nbe;
   r=grip:dr:Reff-dr;,r=r+dr/2;
end
```

```
   if NL_TWIST_VAL==1
      NL_TWIST=NL_TWIST/57.3;
      n=length(NL_TWIST);
      if RADSPC_VAL==1
         y=((Reff/R)-NEW_r(n))*((NL_TWIST(n)-NL_TWIST(n-1))/(NEW_r(n)-
NEW_r(n-1)));
      else
         y=((Reff/R)-r(n))*((NL_TWIST(n)-NL_TWIST(n-1))/(r(n)- r(n-1)));
      end
      NL_TWIST1=[NL_TWIST (NL_TWIST(n)+y)];
      m=length(NL_TWIST1);
      dTW=diff(NL_TWIST1);
      twist=(NL_TWIST1(1:m-1))+dTW/2;
      betat=twist;
   else
      betat=twist*(0.7-(r/R));

   end
   rT1=rT2;,% *** Set value for  rT as calculated for tirm ***

   RbarT=rT1*Rbar;

   mblade=wblade/32.17;


   betao=betao-(theta_inc*(betao/theta_save));
   psi=0:360/naz:360-360/naz;,psi=psi'/57.3;



   %% set up vector of blade element chords and then varies them as

   %%  requested with the blade taper and blade taper start position

   %%  rchord=root chord

   %%  cblade=vector of blade element chord lengths

   %%  tr=taper ratio (tip/root)

   %%  trst=taper ratio start position (r/R)


   cblade=rchord*ones(size(r));     % gives all elements same chord
length initially


   if tr==0 % prevents division by zero later in code

      tr=1; %  in case 0 is enter for taper ratio instead

   end            %  of 1 for no taper
```

```
   if trst==0
      slope=(rchord-rchord*tr)/(Reff-grip);     % Modifies each element
      cblade=cblade-slope*(r-grip);       %  chord length wrt input
      tchord=cblade(nbe);                       %  taper ratio which has
been
      mchord=sum(cblade)/nbe;             %  been converted into a slope
      %  top portion takes into
   else                                   %  account the possibility
that
      slope=(rchord-rchord*tr)/(R*(1-trst));    %  a 0 start position is
really at
      z=fix(nbe*trst);                    %  the start of the aero
portion
      if z<=1           % prevents beginning index fm being zero
         z=1;
      end
      cblade(z:nbe)=cblade(z:nbe)-(r(z:nbe)-r(z))*slope;
      tchord=cblade(nbe);
      mchord=sum(cblade)/nbe;
   end


%   *** induced velocity determination ***

   vi=lamdaT*Vtip-Vinf*sin(alphaT);
   vi=vi*ones(size(r));


%   *** Calculate theta based on trim conditions ***
```

```
   theta=thetao+theta1c.*cos(psi)+theta1s.*sin(psi);




   %   *** rotor trimming routine ***


   set(H_STATUS,'String','CALULATING ')
   set(H_STATUS1,'String',['RUN ELAPSED TIME IS ' num2str(fix(toc)) '
SECONDS'])
   pause(3)
   set(H_STATUS2,'String','')


   Tpsi=zeros(size(psi));
   Npsi=zeros(size(psi));
   thrcalc

   Trotor=mean(Tpsi)*b;

   Mpsi(:,k)=zeros(size(psi));

   tmcalc



   % *** calculating drag moments ***
   set(H_STATUS2,'String','CALCULATING DRAG MOMENT')
   set(H_STATUS1,'String',['RUN ELAPSED TIME IS ' num2str(fix(toc)) '
SECONDS'])
   pause(2)
   DMpsi=zeros(size(psi));

   dmcalc
   Qrotor=mean(DMpsi)*b;

   %   *** calculating rotor H force ***

   set(H_STATUS2,'String','CALCULATING ROTOR DRAG')
   set(H_STATUS1,'String',['RUN ELAPSED TIME IS ' num2str(fix(toc)) '
SECONDS'])
   pause(2)


   dT=[dT ddT];
   dN=[dN ddN];
   dD=[dD ddD];

   for i=1:length(r)+1,
```

```
      H1c(i)=2*sum(dT(:,i).*cos(psi))/naz;

      H1s(i)=2*sum(dD(:,i).*sin(psi))/naz;

   end

   Hrotor=(((b*cos(alphaT)/2)*(sum(H1s)-sin(betao)*sum(H1c)))+Drotor)/2;



   CT=Trotor/(Adisk*rho*Vtip^2);
   CH=Hrotor/(Adisk*rho*Vtip^2);
   CQ=Qrotor/(Adisk*rho*Vtip^2*R);


   solidity=b*(sum(cblade)/length(r))/(pi*R);

   vctsig_to(1,n)=CT/solidity;
   vcqsig_to(1,n)=CQ/solidity;
   vchsig_to(1,n)=CH/solidity;
   vmu_to(1,n)=mu;



   %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%NOT sure after
here*****************************


   % assume alpha of shaft is zero

   lockno=rho*a*(sum(cblade)/length(r))*R^4/Ib;

   thetao=thetao-.7*theta1;

    A1=-theta1c;
    B1=-theta1s;

    ctsig=vctsig_to(1,n);

    altpp=-alphaT;

   num1=thetao*mu*8/3;
   num2=2*theta1*mu;
   num3=-B1*(1+((3/2)*mu^2));
   num4=2*mu*(mu*altpp-(ctsig*solidity/2/mu));
   den1=1-((mu^2)/2);
   term1=(num1+num2+num3)/den1;

   num5=12*(e/R);
   den5=lockno*((1-(e/R))^3)*(1+(mu^4)/4);
   term2=num5/den5;

   num6=A1*(1+(.5*mu^2));
   num7=(4/3)*ctsig;
   num8=(2/3)*mu*lockno/a;
   num9=solidity/2*mu;
   den6=1+(3*e/2/R);
```

```
   term3=num6+(num7*((num8/den6)+num9));

   va1s_to(1,n)=(term1+(term2*term3));

   vvi_to(1,n)=mean(vi);

   bnum1=(4/3)*ctsig;
   bden1=1+((mu^2)/2);
   bnum2=(2/3)*mu*lockno/a;
   bden2=1+(3*e/2/R);
   bterm1=solidity/2/mu;
   bterm2=A1+((bnum1/bden1)*((bnum2/bden2)+bterm1));
   bnum3=12*e/R;
   bden3=lockno*((1-(e/R))^3)*(1-(.25*(mu^4)));
   bterm3=bnum3/bden3;
   bnum4=2*mu*((mu*(altpp-altpp))-(ctsig*solidity/2/mu));

   bterm4=num1+num2+num3+bnum4;

   vb1s_to(1,n)=bterm2+(bterm3*bterm4);

   %tail rotor
   %
   if tailrot==1
      ctsig=vctsig_to(1,n);
      cqsig=vcqsig_to(1,n);
      ohm=omega;
      lv=lvd-xcg;
      g=32.2;
      A=pi*R*R;


      Abt=bt*Rt*ct;
      At=pi*Rt*Rt;
      mut=Vinf/ohmt/Rt;
      sigmat=Abt/At;
      lt=ltd-xcg;
      ht=htd-zcg;
      yt=ytd-ycg;
      locknot=rho*at*ct*Rt^4/Ibt;
      Tt=(cqsig*solidity*rho*A*(ohm*R)^2*R-Lvert*lv)/lt;
      ctsigt=Tt/(rho*At*(ohmt*Rt)^2);
      lampt=-ctsigt*solidity/2/mu;
      aot=2/3*locknot*ctsigt/at-3/2*g*Rt^2/(ohmt*Rt)^2;
      a1st=(-2*((4/3*mut*aot+ctsigt*sigmat/2/mut)/(2+mut*mut))*(1-
(4*mut/(2+3*mut*mut))^2)*...
         tan(delta3))/(((2-mut*mut)/(2+3*mut*mut))+(1-
(4*mut/(2+3*mut*mut))^2)*tan(delta3)^2);

b1st=2*((4/3*mu*aot*ctsigt*solidity/2/mu)/(2+mu*mu))+a1st*tan(delta3);
      thetaot=((4*ctsigt/at)-
(.5+mut*mut/2)*theta1t+mut*b1st*tan(delta3)-lampt)/...
         (2/3+mut*mut)-aot*tan(delta3);
      theta75t=thetaot+.75*theta1t;
      v1t=sqrt(Tt/2/rho/At);
      vctsigt_tot(1,n)=ctsigt;
      vthetaot(1,n)=thetaot;
```

```
    end

end
```

23. stab_control_input_fcn.m

```
function stab_control_input_fcn(Action)

%     Switchyard Callback function for stab_control_input_1.m and
stab_control_input_2.m
%     Written for JANRAD version 6.0 by LT David A. Heathorn

global  S_STAB_INPUT_1 S_STAB_INPUT_2 H_STAB_IN S_PERF_INPUT
S_PERF_OUTPUT H_STAB_IN1...
  H_STAB_IN2 S_SC_INPUT_1 S_SC_INPUT_2

if isempty(nargin)
   return
end

   switch Action
   case 'cont1'
     if isempty(getfield(S_SC_INPUT_1,'Ib')|...
           getfield(S_SC_INPUT_1,'hmd')|...
            getfield(S_SC_INPUT_1,'lmd')|...
            getfield(S_SC_INPUT_1,'ymd')|...
            getfield(S_SC_INPUT_1,'im')|...
            getfield(S_SC_INPUT_1,'hvd')|...
            getfield(S_SC_INPUT_1,'lvd')|...
            getfield(S_SC_INPUT_1,'yvd')|...
            getfield(S_SC_INPUT_1,'alplov')|...
            getfield(S_SC_INPUT_1,'clvertmax')|...
            getfield(S_SC_INPUT_1,'av')|...
            getfield(S_SC_INPUT_1,'hhd')|...
            getfield(S_SC_INPUT_1,'lhd')|...
            getfield(S_SC_INPUT_1,'alploh')|...
            getfield(S_SC_INPUT_1,'ih')|...
            getfield(S_SC_INPUT_1,'ah')|...
            getfield(S_SC_INPUT_1,'qhq')|...
            getfield(S_SC_INPUT_1,'vhv1')|...
            getfield(S_SC_INPUT_1,'detafdalpfh')|...
            getfield(S_SC_INPUT_1,'htd')|...
            getfield(S_SC_INPUT_1,'ytd')|...
            getfield(S_SC_INPUT_1,'bt')|...
            getfield(S_SC_INPUT_1,'ct')|...
            getfield(S_SC_INPUT_1,'Rt')|...
            getfield(S_SC_INPUT_1,'at')|...
            getfield(S_SC_INPUT_1,'ohmt')|...
            getfield(S_SC_INPUT_1,'Ibt')|...
            getfield(S_SC_INPUT_1,'delta3')|...
            getfield(S_SC_INPUT_1,'theta1t'));

         empty_boxes
        return
      end

      stability_control_input_2
      close(H_STAB_IN1)

   case 'cont2'
      if isempty(getfield(S_SC_INPUT_2,'hwd')|...
```

```
            getfield(S_SC_INPUT_2,'lwd')|...
            getfield(S_SC_INPUT_2,'ywd')|...
            getfield(S_SC_INPUT_2,'alplow')|...
            getfield(S_SC_INPUT_2,'iw')|...
            getfield(S_SC_INPUT_2,'aw')|...
            getfield(S_SC_INPUT_2,'ctw')|...
            getfield(S_SC_INPUT_2,'crw')|...
            getfield(S_SC_INPUT_2,'vwv1')|...
            getfield(S_SC_INPUT_2,'detafdalpfw')|...
            getfield(S_SC_INPUT_2,'zcg')|...
            getfield(S_SC_INPUT_2,'xcg')|...
            getfield(S_SC_INPUT_2,'ycg')|...
            getfield(S_SC_INPUT_2,'Ixx')|...
            getfield(S_SC_INPUT_2,'Iyy')|...
            getfield(S_SC_INPUT_2,'Izz')|...
            getfield(S_SC_INPUT_2,'Ixz')|...
            getfield(S_SC_INPUT_2,'vfv1')|...
            getfield(S_SC_INPUT_2,'htnd')|...
            getfield(S_SC_INPUT_2,'ltnd')|...
            getfield(S_SC_INPUT_2,'ytnd')|...
            getfield(S_SC_INPUT_2,'dian')|...
            getfield(S_SC_INPUT_2,'swirl')|...
            getfield(S_SC_INPUT_2,'Ytmaxn')|...
            getfield(S_SC_INPUT_2,'lttnd')|...
            getfield(S_SC_INPUT_2,'db1mddele')|...
            getfield(S_SC_INPUT_2,'da1mddela')|...
            getfield(S_SC_INPUT_2,'dthetomddelc')|...
            getfield(S_SC_INPUT_2,'dthetotddelp')|...
            getfield(S_SC_INPUT_2,'sidearm')|...
            getfield(S_SC_INPUT_2,'maxr'));

         empty_boxes
      end
      close(H_STAB_IN2)
      sc_status

   case 'cnx'
      analysis
      close (gcf)
   case 'back1'
      options
      close (H_PERF_IN1)
   case 'back2'
      stability_control_input_1
      close (H_PERF_IN2)
   case 'print'
      set(gcf,'PaperOrientation','landscape')
      set(gcf,'PaperPosition',[.5 .5 10 7.5])
      print -dwinc
   case 'return'
      janrad98
      close all
   case 'quit'
      quit_gui
   case 'about'
      about_janrad
   case 'mesh'
```

```
   airfoil_mesh
case 'ok'
   close (H_AF_MESH)
end
```

24. stab_out_1.m

```
function stab_out_1()
% This is the machine-generated representation of a MATLAB object
% and its children.  Note that handle values may change when these
% objects are re-created. This may cause problems with some callbacks.
% The command syntax may be supported in the future, but is currently
% incomplete and subject to change.
%
% To re-open this system, just type the name of the m-file at the MATLAB
% prompt. The M-file and its associtated MAT-file must be on your path.

% Created for JANRAD version 6.0 by LT David A. Heathorn

load stab_out_1
global Amat Bmat H_STAB_OUT
load mat_temp
A=Amat(1:8,1:8);
B=Bmat(1:8,1:4);


H_STAB_OUT = figure('Color',[0.8 0.8 0.8], ...
      'Colormap',mat0, ...
   'PointerShapeCData',mat1, ...
   'Name','Linear Model of Helicopter', ...
   'Position',[2 28 798 534], ...
   'Units','points',...
   'NumberTitle','off', ...
   'Tag','Fig1');
b = uicontrol('Parent',H_STAB_OUT, ...
      'Units','points', ...
      'Position',[2.25 189.75 435.75 207.75], ...
      'Style','frame', ...
      'Tag','Frame1');
b = uicontrol('Parent',H_STAB_OUT, ...
      'Units','points', ...
      'Position',[3 0.75 434.25 183.75], ...
      'Style','frame', ...
   'Tag','Frame2');
b = uicontrol('Parent',H_STAB_OUT, ...
      'Units','points', ...
      'BackgroundColor',[0.752941 0.752941 0.752941], ...
      'HorizontalAlignment','left', ...
      'Position',[443.25 128.25 52.5 255.75], ...
      'Style','frame', ...
      'Tag','Frame3');
b = uicontrol('Parent',H_STAB_OUT, ...
      'Units','points', ...
      'BackgroundColor',[0.752941 0.752941 0.752941], ...
      'Position',[500.25 237.75 93.75 145.5], ...
      'Style','frame', ...
      'Tag','Frame4');
b = uimenu('Parent',H_STAB_OUT, ...
      'Label','JANRAD Options', ...
      'Tag','uimenu1');
c = uimenu('Parent',b, ...
      'Callback','stab_out_1_fcn quit', ...
```

```
        'Label','Quit JANRAD', ...
        'Tag','JANRAD OptionsSubuimenu1');
c = uimenu('Parent',b, ...
        'Callback','stab_out_1_fcn return', ...
        'Label','Return to Begining', ...
        'Tag','JANRAD OptionsSubuimenu1');
c = uimenu('Parent',b, ...
        'Callback','stab_out_1_fcn delta_input', ...
        'Enable','off', ...
        'Label','Change Input Parameters', ...
        'Tag','Subuimenu1');
c = uimenu('Parent',b, ...
        'Callback','performance_input_fcn about', ...
        'Label','About Janrad 98 ...', ...
        'Separator','on', ...
        'Tag','Subuimenu1');

b = uicontrol('Parent',H_STAB_OUT, ...
        'Units','points', ...
        'Position',[465.75 55.5 116.25 23.25], ...
    'String','Save to File', ...
    'Callback','stab_out_1_fcn save',...
        'Tag','Pushbutton1');
b = uicontrol('Parent',H_STAB_OUT, ...
        'Units','points', ...
        'BackgroundColor',[0.752941 0.752941 0.752941], ...
        'FontSize',18, ...
        'Position',[58.5 375.75 322.5 18.75], ...
        'String','A Matrix of the form xdot=Ax+Bu', ...
        'Style','text', ...
        'Tag','StaticText2');
b = uicontrol('Parent',H_STAB_OUT, ...
        'Units','points', ...
        'BackgroundColor',[0.752941 0.752941 0.752941], ...
        'FontSize',18, ...
        'Position',[85.5 92.25 92.25 18.75], ...
        'String','B Matrix', ...
        'Style','text', ...
        'Tag','StaticText2');
b = uicontrol('Parent',H_STAB_OUT, ...
        'Units','points', ...
        'Position',[466.5 94.5 114.75 23.25], ...
    'String','Coupled Response', ...
    'Callback','stab_out_1_fcn cr',...
        'Tag','Pushbutton1');
b = uicontrol('Parent',H_STAB_OUT, ...
        'Units','points', ...
        'BackgroundColor',[0.752941 0.752941 0.752941], ...
        'FontName','arial', ...
        'FontSize',7, ...
        'Position',[12 355.5 45 15], ...
        'String',A(1,1), ...
        'Style','text', ...
        'Tag','StaticText1');
b = uicontrol('Parent',H_STAB_OUT, ...
        'Units','points', ...
        'BackgroundColor',[0.752941 0.752941 0.752941], ...
```

```
        'FontName','arial', ...
        'FontSize',7, ...
        'Position',[66 356.25 45 15], ...
        'String',A(1,2), ...
        'Style','text', ...
        'Tag','StaticText1');
b = uicontrol('Parent',H_STAB_OUT, ...
        'Units','points', ...
        'FontName','arial', ...
        'FontSize',7, ...
        'Position',[170.25 356.25 45 15], ...
        'String',A(1,4), ...
        'Style','text', ...
        'Tag','StaticText1');
b = uicontrol('Parent',H_STAB_OUT, ...
        'Units','points', ...
        'FontName','arial', ...
        'FontSize',7, ...
        'Position',[118.5 356.25 45 15], ...
        'String',A(1,3), ...
        'Style','text', ...
        'Tag','StaticText1');
b = uicontrol('Parent',H_STAB_OUT, ...
        'Units','points', ...
        'FontName','arial', ...
        'FontSize',7, ...
        'Position',[327.75 356.25 45 15], ...
        'String',A(1,7), ...
        'Style','text', ...
        'Tag','StaticText1');
b = uicontrol('Parent',H_STAB_OUT, ...
        'Units','points', ...
        'FontName','arial', ...
        'FontSize',7, ...
        'Position',[379.5 356.25 45 15], ...
        'String',A(1,8), ...
        'Style','text', ...
        'Tag','StaticText1');
b = uicontrol('Parent',H_STAB_OUT, ...
        'Units','points', ...
        'FontName','arial', ...
        'FontSize',7, ...
        'Position',[275.25 356.25 45 15], ...
        'String',A(1,6), ...
        'Style','text', ...
        'Tag','StaticText1');
b = uicontrol('Parent',H_STAB_OUT, ...
        'Units','points', ...
        'FontName','arial', ...
        'FontSize',7, ...
        'Position',[222 356.25 45 15], ...
        'String',A(1,5), ...
        'Style','text', ...
        'Tag','StaticText1');
b = uicontrol('Parent',H_STAB_OUT, ...
        'Units','points', ...
        'FontName','arial', ...
```

```
        'FontSize',7, ...
        'Position',[222.75 330.75 45 15], ...
        'String',A(2,5), ...
        'Style','text', ...
        'Tag','StaticText1');
b = uicontrol('Parent',H_STAB_OUT, ...
        'Units','points', ...
        'FontName','arial', ...
        'FontSize',7, ...
        'Position',[277.5 330.75 45 15], ...
        'String',A(2,6), ...
        'Style','text', ...
        'Tag','StaticText1');
b = uicontrol('Parent',H_STAB_OUT, ...
        'Units','points', ...
        'FontName','arial', ...
        'FontSize',7, ...
        'Position',[379.5 330.75 45 15], ...
        'String',A(2,8), ...
        'Style','text', ...
        'Tag','StaticText1');
b = uicontrol('Parent',H_STAB_OUT, ...
        'Units','points', ...
        'FontName','arial', ...
        'FontSize',7, ...
        'Position',[327.75 330.75 45 15], ...
        'String',A(2,7), ...
        'Style','text', ...
        'Tag','StaticText1');
b = uicontrol('Parent',H_STAB_OUT, ...
        'Units','points', ...
        'FontName','arial', ...
        'FontSize',7, ...
        'Position',[118.5 330.75 45 15], ...
        'String',A(2,3), ...
        'Style','text', ...
        'Tag','StaticText1');
b = uicontrol('Parent',H_STAB_OUT, ...
        'Units','points', ...
        'FontName','arial', ...
        'FontSize',7, ...
        'Position',[170.25 330.75 45 15], ...
        'String',A(2,4), ...
        'Style','text', ...
        'Tag','StaticText1');
b = uicontrol('Parent',H_STAB_OUT, ...
        'Units','points', ...
        'FontName','arial', ...
        'FontSize',7, ...
        'Position',[66 330.75 45 15], ...
        'String',A(2,2), ...
        'Style','text', ...
        'Tag','StaticText1');
b = uicontrol('Parent',H_STAB_OUT, ...
        'Units','points', ...
        'FontName','arial', ...
        'FontSize',7, ...
```

```
        'Position',[13.5 330 45 15], ...
        'String',A(2,1), ...
        'Style','text', ...
        'Tag','StaticText1');
b = uicontrol('Parent',H_STAB_OUT, ...
        'Units','points', ...
        'FontName','arial', ...
        'FontSize',7, ...
        'Position',[13.5 307.5 45 15], ...
        'String',A(3,1), ...
        'Style','text', ...
        'Tag','StaticText1');
b = uicontrol('Parent',H_STAB_OUT, ...
        'Units','points', ...
        'FontName','arial', ...
        'FontSize',7, ...
        'Position',[65.25 307.5 45 15], ...
        'String',A(3,2), ...
        'Style','text', ...
        'Tag','StaticText1');
b = uicontrol('Parent',H_STAB_OUT, ...
        'Units','points', ...
        'FontName','arial', ...
        'FontSize',7, ...
        'Position',[170.25 307.5 45 15], ...
        'String',A(3,4), ...
        'Style','text', ...
        'Tag','StaticText1');
b = uicontrol('Parent',H_STAB_OUT, ...
        'Units','points', ...
        'FontName','arial', ...
        'FontSize',7, ...
        'Position',[118.5 307.5 45 15], ...
        'String',A(3,3), ...
        'Style','text', ...
        'Tag','StaticText1');
b = uicontrol('Parent',H_STAB_OUT, ...
        'Units','points', ...
        'FontName','arial', ...
        'FontSize',7, ...
        'Position',[327.75 307.5 45 15], ...
        'String',A(3,7), ...
        'Style','text', ...
        'Tag','StaticText1');
b = uicontrol('Parent',H_STAB_OUT, ...
        'Units','points', ...
        'FontName','arial', ...
        'FontSize',7, ...
        'Position',[379.5 307.5 45 15], ...
        'String',A(3,8), ...
        'Style','text', ...
        'Tag','StaticText1');
b = uicontrol('Parent',H_STAB_OUT, ...
        'Units','points', ...
        'FontName','arial', ...
        'FontSize',7, ...
        'Position',[275.25 307.5 45 15], ...
```

```
        'String',A(3,6), ...
        'Style','text', ...
        'Tag','StaticText1');
b = uicontrol('Parent',H_STAB_OUT, ...
        'Units','points', ...
        'FontName','arial', ...
        'FontSize',7, ...
        'Position',[222.75 307.5 45 15], ...
        'String',A(3,5), ...
        'Style','text', ...
        'Tag','StaticText1');
b = uicontrol('Parent',H_STAB_OUT, ...
        'Units','points', ...
        'FontName','arial', ...
        'FontSize',7, ...
        'Position',[222.75 284.25 45 15], ...
        'String',A(4,5), ...
        'Style','text', ...
        'Tag','StaticText1');
b = uicontrol('Parent',H_STAB_OUT, ...
        'Units','points', ...
        'FontName','arial', ...
        'FontSize',7, ...
        'Position',[275.25 284.25 45 15], ...
        'String',A(4,6), ...
        'Style','text', ...
        'Tag','StaticText1');
b = uicontrol('Parent',H_STAB_OUT, ...
        'Units','points', ...
        'FontName','arial', ...
        'FontSize',7, ...
        'Position',[379.5 284.25 45 15], ...
        'String',A(4,8), ...
        'Style','text', ...
        'Tag','StaticText1');
b = uicontrol('Parent',H_STAB_OUT, ...
        'Units','points', ...
        'FontName','arial', ...
        'FontSize',7, ...
        'Position',[327.75 284.25 45 15], ...
        'String',A(4,7), ...
        'Style','text', ...
        'Tag','StaticText1');
b = uicontrol('Parent',H_STAB_OUT, ...
        'Units','points', ...
        'FontName','arial', ...
        'FontSize',7, ...
        'Position',[118.5 284.25 45 15], ...
        'String',A(4,3), ...
        'Style','text', ...
        'Tag','StaticText1');
b = uicontrol('Parent',H_STAB_OUT, ...
        'Units','points', ...
        'FontName','arial', ...
        'FontSize',7, ...
        'Position',[170.25 284.25 45 15], ...
        'String',A(4,4), ...
```

```
        'Style','text', ...
        'Tag','StaticText1');
b = uicontrol('Parent',H_STAB_OUT, ...
        'Units','points', ...
        'FontName','arial', ...
        'FontSize',7, ...
        'Position',[66 284.25 45 15], ...
        'String',A(4,2), ...
        'Style','text', ...
        'Tag','StaticText1');
b = uicontrol('Parent',H_STAB_OUT, ...
        'Units','points', ...
        'FontName','arial', ...
        'FontSize',7, ...
        'Position',[13.5 284.25 45 15], ...
        'String',A(4,1), ...
        'Style','text', ...
        'Tag','StaticText1');
b = uicontrol('Parent',H_STAB_OUT, ...
        'Units','points', ...
        'FontName','arial', ...
        'FontSize',7, ...
        'Position',[13.5 261.75 45 15], ...
        'String',A(5,1), ...
        'Style','text', ...
        'Tag','StaticText1');
b = uicontrol('Parent',H_STAB_OUT, ...
        'Units','points', ...
        'FontName','arial', ...
        'FontSize',7, ...
        'Position',[66 261.75 45 15], ...
        'String',A(5,2), ...
        'Style','text', ...
        'Tag','StaticText1');
b = uicontrol('Parent',H_STAB_OUT, ...
        'Units','points', ...
        'FontName','arial', ...
        'FontSize',7, ...
        'Position',[170.25 261.75 45 15], ...
        'String',A(5,4), ...
        'Style','text', ...
        'Tag','StaticText1');
b = uicontrol('Parent',H_STAB_OUT, ...
        'Units','points', ...
        'FontName','arial', ...
        'FontSize',7, ...
        'Position',[118.5 261.75 45 15], ...
        'String',A(5,3), ...
        'Style','text', ...
        'Tag','StaticText1');
b = uicontrol('Parent',H_STAB_OUT, ...
        'Units','points', ...
        'FontName','arial', ...
        'FontSize',7, ...
        'Position',[327.75 261.75 45 15], ...
        'String',A(5,7), ...
        'Style','text', ...
```

```
      'Tag','StaticText1');
b = uicontrol('Parent',H_STAB_OUT, ...
      'Units','points', ...
      'FontName','arial', ...
      'FontSize',7, ...
      'Position',[379.5 261.75 45 15], ...
      'String',A(5,8), ...
      'Style','text', ...
      'Tag','StaticText1');
b = uicontrol('Parent',H_STAB_OUT, ...
      'Units','points', ...
      'FontName','arial', ...
      'FontSize',7, ...
      'Position',[275.25 261.75 45 15], ...
      'String',A(5,6), ...
      'Style','text', ...
      'Tag','StaticText1');
b = uicontrol('Parent',H_STAB_OUT, ...
      'Units','points', ...
      'FontName','arial', ...
      'FontSize',7, ...
      'Position',[222.75 261.75 45 15], ...
      'String',A(5,5), ...
      'Style','text', ...
      'Tag','StaticText1');
b = uicontrol('Parent',H_STAB_OUT, ...
      'Units','points', ...
      'FontName','arial', ...
      'FontSize',7, ...
      'Position',[222.75 237.75 45 15], ...
      'String',A(6,5), ...
      'Style','text', ...
      'Tag','StaticText1');
b = uicontrol('Parent',H_STAB_OUT, ...
      'Units','points', ...
      'FontName','arial', ...
      'FontSize',7, ...
      'Position',[275.25 237.75 45 15], ...
      'String',A(6,6), ...
      'Style','text', ...
      'Tag','StaticText1');
b = uicontrol('Parent',H_STAB_OUT, ...
      'Units','points', ...
      'BackgroundColor',[0.752941 0.752941 0.752941], ...
      'FontName','arial', ...
      'FontSize',7, ...
      'Position',[379.5 237.75 45 15], ...
      'String',A(6,8), ...
      'Style','text', ...
      'Tag','StaticText1');
b = uicontrol('Parent',H_STAB_OUT, ...
      'Units','points', ...
      'FontName','arial', ...
      'FontSize',7, ...
      'Position',[327.75 237.75 45 15], ...
      'String',A(6,7), ...
      'Style','text', ...
```

```
        'Tag','StaticText1');
b = uicontrol('Parent',H_STAB_OUT, ...
        'Units','points', ...
        'FontName','arial', ...
        'FontSize',7, ...
        'Position',[118.5 237.75 45 15], ...
        'String',A(6,3), ...
        'Style','text', ...
        'Tag','StaticText1');
b = uicontrol('Parent',H_STAB_OUT, ...
        'Units','points', ...
        'FontName','arial', ...
        'FontSize',7, ...
        'Position',[170.25 237.75 45 15], ...
        'String',A(6,4), ...
        'Style','text', ...
        'Tag','StaticText1');
b = uicontrol('Parent',H_STAB_OUT, ...
        'Units','points', ...
        'FontName','arial', ...
        'FontSize',7, ...
        'Position',[66 237.75 45 15], ...
        'String',A(6,2), ...
        'Style','text', ...
        'Tag','StaticText1');
b = uicontrol('Parent',H_STAB_OUT, ...
        'Units','points', ...
        'FontName','arial', ...
        'FontSize',7, ...
        'Position',[13.5 237.75 45 15], ...
        'String',A(6,1), ...
        'Style','text', ...
        'Tag','StaticText1');
b = uicontrol('Parent',H_STAB_OUT, ...
        'Units','points', ...
        'FontName','arial', ...
        'FontSize',7, ...
        'Position',[13.5 215.25 45 15], ...
        'String',A(7,1), ...
        'Style','text', ...
        'Tag','StaticText1');
b = uicontrol('Parent',H_STAB_OUT, ...
        'Units','points', ...
        'FontName','arial', ...
        'FontSize',7, ...
        'Position',[66 215.25 45 15], ...
        'String',A(7,2), ...
        'Style','text', ...
        'Tag','StaticText1');
b = uicontrol('Parent',H_STAB_OUT, ...
        'Units','points', ...
        'BackgroundColor',[0.752941 0.752941 0.752941], ...
        'FontName','arial', ...
        'FontSize',7, ...
        'Position',[170.25 215.25 45 15], ...
        'String',A(7,4), ...
        'Style','text', ...
```

```
        'Tag','StaticText1');
b = uicontrol('Parent',H_STAB_OUT, ...
        'Units','points', ...
        'FontName','arial', ...
        'FontSize',7, ...
        'Position',[118.5 215.25 45 15], ...
        'String',A(7,3), ...
        'Style','text', ...
        'Tag','StaticText1');
b = uicontrol('Parent',H_STAB_OUT, ...
        'Units','points', ...
        'FontName','arial', ...
        'FontSize',7, ...
        'Position',[327.75 215.25 45 15], ...
        'String',A(7,7), ...
        'Style','text', ...
        'Tag','StaticText1');
b = uicontrol('Parent',H_STAB_OUT, ...
        'Units','points', ...
        'FontName','arial', ...
        'FontSize',7, ...
        'Position',[379.5 215.25 45 15], ...
        'String',A(7,8), ...
        'Style','text', ...
        'Tag','StaticText1');
b = uicontrol('Parent',H_STAB_OUT, ...
        'Units','points', ...
        'BackgroundColor',[0.752941 0.752941 0.752941], ...
        'FontName','arial', ...
        'FontSize',7, ...
        'Position',[275.25 215.25 45 15], ...
        'String',A(7,6), ...
        'Style','text', ...
        'Tag','StaticText1');
b = uicontrol('Parent',H_STAB_OUT, ...
        'Units','points', ...
        'FontName','arial', ...
        'FontSize',7, ...
        'Position',[222.75 215.25 45 15], ...
        'String',A(7,5), ...
        'Style','text', ...
        'Tag','StaticText1');
b = uicontrol('Parent',H_STAB_OUT, ...
        'Units','points', ...
        'FontName','arial', ...
        'FontSize',7, ...
        'Position',[222.75 194.25 45 15], ...
        'String',A(8,5), ...
        'Style','text', ...
        'Tag','StaticText1');
b = uicontrol('Parent',H_STAB_OUT, ...
        'Units','points', ...
        'FontName','arial', ...
        'FontSize',7, ...
        'Position',[275.25 193.5 45 15], ...
        'String',A(8,6), ...
        'Style','text', ...
```

```
        'Tag','StaticText1');
b = uicontrol('Parent',H_STAB_OUT, ...
        'Units','points', ...
        'FontName','arial', ...
        'FontSize',7, ...
        'Position',[379.5 193.5 45 15], ...
        'String',A(8,8), ...
        'Style','text', ...
        'Tag','StaticText1');
b = uicontrol('Parent',H_STAB_OUT, ...
        'Units','points', ...
        'FontName','arial', ...
        'FontSize',7, ...
        'Position',[327.75 193.5 45 15], ...
        'String',A(8,7), ...
        'Style','text', ...
        'Tag','StaticText1');
b = uicontrol('Parent',H_STAB_OUT, ...
        'Units','points', ...
        'FontName','arial', ...
        'FontSize',7, ...
        'Position',[118.5 193.5 45 15], ...
        'String',A(8,3), ...
        'Style','text', ...
        'Tag','StaticText1');
b = uicontrol('Parent',H_STAB_OUT, ...
        'Units','points', ...
        'FontName','arial', ...
        'FontSize',7, ...
        'Position',[170.25 193.5 45 15], ...
        'String',A(8,4), ...
        'Style','text', ...
        'Tag','StaticText1');
b = uicontrol('Parent',H_STAB_OUT, ...
        'Units','points', ...
        'FontName','arial', ...
        'FontSize',7, ...
        'Position',[66 193.5 45 15], ...
        'String',A(8,2), ...
        'Style','text', ...
        'Tag','StaticText1');
b = uicontrol('Parent',H_STAB_OUT, ...
        'Units','points', ...
        'BackgroundColor',[0.752941 0.752941 0.752941], ...
        'FontName','arial', ...
        'FontSize',7, ...
        'Position',[13.5 192.75 45 15], ...
        'String',A(8,1), ...
        'Style','text', ...
        'Tag','StaticText1');
b = uicontrol('Parent',H_STAB_OUT, ...
        'Units','points', ...
        'Position',[466.5 14.25 114.75 23.25], ...
    'String','Modify Inputs and Retrim', ...
    'Callback','stab_out_1_fcn modin',...
        'Tag','Pushbutton1');
b = uicontrol('Parent',H_STAB_OUT, ...
```

```
        'Units','points', ...
        'BackgroundColor',[0.752941 0.752941 0.752941], ...
        'FontName','arial', ...
        'FontSize',7, ...
        'Position',[220.5 3 45 15], ...
        'String',B(8,1), ...
        'Style','text', ...
        'Tag','StaticText1');
b = uicontrol('Parent',H_STAB_OUT, ...
        'Units','points', ...
        'FontName','arial', ...
        'FontSize',7, ...
        'Position',[273 3.75 45 15], ...
        'String',B(8,2), ...
        'Style','text', ...
        'Tag','StaticText1');
b = uicontrol('Parent',H_STAB_OUT, ...
        'Units','points', ...
        'FontName','arial', ...
        'FontSize',7, ...
        'Position',[377.25 3.75 45 15], ...
        'String',B(8,4), ...
        'Style','text', ...
        'Tag','StaticText1');
b = uicontrol('Parent',H_STAB_OUT, ...
        'Units','points', ...
        'FontName','arial', ...
        'FontSize',7, ...
        'Position',[325.5 3.75 45 15], ...
        'String',B(8,3), ...
        'Style','text', ...
        'Tag','StaticText1');
b = uicontrol('Parent',H_STAB_OUT, ...
        'Units','points', ...
        'FontName','arial', ...
        'FontSize',7, ...
        'Position',[325.5 25.5 45 15], ...
        'String',B(7,3), ...
        'Style','text', ...
        'Tag','StaticText1');
b = uicontrol('Parent',H_STAB_OUT, ...
        'Units','points', ...
        'BackgroundColor',[0.752941 0.752941 0.752941], ...
        'FontName','arial', ...
        'FontSize',7, ...
        'Position',[377.25 25.5 45 15], ...
        'String',B(7,4), ...
        'Style','text', ...
        'Tag','StaticText1');
b = uicontrol('Parent',H_STAB_OUT, ...
        'Units','points', ...
        'FontName','arial', ...
        'FontSize',7, ...
        'Position',[273 25.5 45 15], ...
        'String',B(7,2), ...
        'Style','text', ...
        'Tag','StaticText1');
```

```
b = uicontrol('Parent',H_STAB_OUT, ...
      'Units','points', ...
      'FontName','arial', ...
      'FontSize',7, ...
      'Position',[220.5 25.5 45 15], ...
      'String',B(7,1), ...
      'Style','text', ...
      'Tag','StaticText1');
b = uicontrol('Parent',H_STAB_OUT, ...
      'Units','points', ...
      'FontName','arial', ...
      'FontSize',7, ...
      'Position',[220.5 48 45 15], ...
      'String',B(6,1), ...
      'Style','text', ...
      'Tag','StaticText1');
b = uicontrol('Parent',H_STAB_OUT, ...
      'Units','points', ...
      'FontName','arial', ...
      'FontSize',7, ...
      'Position',[273 48 45 15], ...
      'String',B(6,2), ...
      'Style','text', ...
      'Tag','StaticText1');
b = uicontrol('Parent',H_STAB_OUT, ...
      'Units','points', ...
      'FontName','arial', ...
      'FontSize',7, ...
      'Position',[377.25 48 45 15], ...
      'String',B(6,4), ...
      'Style','text', ...
      'Tag','StaticText1');
b = uicontrol('Parent',H_STAB_OUT, ...
      'Units','points', ...
      'FontName','arial', ...
      'FontSize',7, ...
      'Position',[325.5 48 45 15], ...
      'String',B(6,3), ...
      'Style','text', ...
      'Tag','StaticText1');
b = uicontrol('Parent',H_STAB_OUT, ...
      'Units','points', ...
      'FontName','arial', ...
      'FontSize',7, ...
      'Position',[325.5 72 45 15], ...
      'String',B(5,3), ...
      'Style','text', ...
      'Tag','StaticText1');
b = uicontrol('Parent',H_STAB_OUT, ...
      'Units','points', ...
      'FontName','arial', ...
      'FontSize',7, ...
      'Position',[377.25 72 45 15], ...
      'String',B(5,4), ...
      'Style','text', ...
      'Tag','StaticText1');
b = uicontrol('Parent',H_STAB_OUT, ...
```

```
        'Units','points', ...
        'FontName','arial', ...
        'FontSize',7, ...
        'Position',[273 72 45 15], ...
        'String',B(5,2), ...
        'Style','text', ...
        'Tag','StaticText1');
b = uicontrol('Parent',H_STAB_OUT, ...
        'Units','points', ...
        'FontName','arial', ...
        'FontSize',7, ...
        'Position',[220.5 72 45 15], ...
        'String',B(5,1), ...
        'Style','text', ...
        'Tag','StaticText1');
b = uicontrol('Parent',H_STAB_OUT, ...
        'Units','points', ...
        'FontName','arial', ...
        'FontSize',7, ...
        'Position',[220.5 94.5 45 15], ...
        'String',B(4,1), ...
        'Style','text', ...
        'Tag','StaticText1');
b = uicontrol('Parent',H_STAB_OUT, ...
        'Units','points', ...
        'FontName','arial', ...
        'FontSize',7, ...
        'Position',[273 94.5 45 15], ...
        'String',B(4,2), ...
        'Style','text', ...
        'Tag','StaticText1');
b = uicontrol('Parent',H_STAB_OUT, ...
        'Units','points', ...
        'FontName','arial', ...
        'FontSize',7, ...
        'Position',[377.25 94.5 45 15], ...
        'String',B(4,4), ...
        'Style','text', ...
        'Tag','StaticText1');
b = uicontrol('Parent',H_STAB_OUT, ...
        'Units','points', ...
        'FontName','arial', ...
        'FontSize',7, ...
        'Position',[325.5 94.5 45 15], ...
        'String',B(4,3), ...
        'Style','text', ...
        'Tag','StaticText1');
b = uicontrol('Parent',H_STAB_OUT, ...
        'Units','points', ...
        'FontName','arial', ...
        'FontSize',7, ...
        'Position',[325.5 117.75 45 15], ...
        'String',B(3,3), ...
        'Style','text', ...
        'Tag','StaticText1');
b = uicontrol('Parent',H_STAB_OUT, ...
        'Units','points', ...
```

```
        'FontName','arial', ...
        'FontSize',7, ...
        'Position',[377.25 117.75 45 15], ...
        'String',B(3,4), ...
        'Style','text', ...
        'Tag','StaticText1');
b = uicontrol('Parent',H_STAB_OUT, ...
        'Units','points', ...
        'FontName','arial', ...
        'FontSize',7, ...
        'Position',[272.25 117.75 45 15], ...
        'String',B(3,2), ...
        'Style','text', ...
        'Tag','StaticText1');
b = uicontrol('Parent',H_STAB_OUT, ...
        'Units','points', ...
        'FontName','arial', ...
        'FontSize',7, ...
        'Position',[220.5 117.75 45 15], ...
        'String',B(3,1), ...
        'Style','text', ...
        'Tag','StaticText1');
b = uicontrol('Parent',H_STAB_OUT, ...
        'Units','points', ...
        'FontName','arial', ...
        'FontSize',7, ...
        'Position',[220.5 140.25 45 15], ...
        'String',B(2,1), ...
        'Style','text', ...
        'Tag','StaticText1');
b = uicontrol('Parent',H_STAB_OUT, ...
        'Units','points', ...
        'FontName','arial', ...
        'FontSize',7, ...
        'Position',[273 141 45 15], ...
        'String',B(2,2), ...
        'Style','text', ...
        'Tag','StaticText1');
b = uicontrol('Parent',H_STAB_OUT, ...
        'Units','points', ...
        'FontName','arial', ...
        'FontSize',7, ...
        'Position',[377.25 141 45 15], ...
        'String',B(2,4), ...
        'Style','text', ...
        'Tag','StaticText1');
b = uicontrol('Parent',H_STAB_OUT, ...
        'Units','points', ...
        'FontName','arial', ...
        'FontSize',7, ...
        'Position',[325.5 141 45 15], ...
        'String',B(2,3), ...
        'Style','text', ...
        'Tag','StaticText1');
b = uicontrol('Parent',H_STAB_OUT, ...
        'Units','points', ...
        'FontName','arial', ...
```

```
      'FontSize',7, ...
      'Position',[325.5 166.5 45 15], ...
      'String',B(1,3), ...
      'Style','text', ...
      'Tag','StaticText1');
b = uicontrol('Parent',H_STAB_OUT, ...
      'Units','points', ...
      'FontName','arial', ...
      'FontSize',7, ...
      'Position',[377.25 166.5 45 15], ...
      'String',B(1,4), ...
      'Style','text', ...
      'Tag','StaticText1');
b = uicontrol('Parent',H_STAB_OUT, ...
      'Units','points', ...
      'BackgroundColor',[0.752941 0.752941 0.752941], ...
      'FontName','arial', ...
      'FontSize',7, ...
      'Position',[273 166.5 45 15], ...
      'String',B(1,2), ...
      'Style','text', ...
      'Tag','StaticText1');
b = uicontrol('Parent',H_STAB_OUT, ...
      'Units','points', ...
      'BackgroundColor',[0.752941 0.752941 0.752941], ...
      'FontName','arial', ...
      'FontSize',7, ...
      'Position',[219 165.75 45 15], ...
      'String',B(1,1), ...
      'Style','text', ...
      'Tag','StaticText1');

%%%%%%%%%%

b = uicontrol('Parent',H_STAB_OUT, ...
      'Units','points', ...
      'BackgroundColor',[0.752941 0.752941 0.752941], ...
      'FontSize',18, ...
      'HorizontalAlignment','left', ...
      'Position',[449.25 348.75 23.25 27.75], ...
      'String','x=', ...
      'Style','text', ...
      'Tag','StaticText3');
b = uicontrol('Parent',H_STAB_OUT, ...
      'Units','points', ...
      'BackgroundColor',[0.752941 0.752941 0.752941], ...
      'FontSize',18, ...
      'HorizontalAlignment','left', ...
      'Position',[508.5 354.75 32.25 22.5], ...
      'String','u=', ...
      'Style','text', ...
      'Tag','StaticText3');
b = uicontrol('Parent',H_STAB_OUT, ...
      'Units','points', ...
      'BackgroundColor',[0.752941 0.752941 0.752941], ...
      'FontSize',18, ...
      'HorizontalAlignment','left', ...
```

```
        'Position',[449.25 295.5 23.25 23.25], ...
        'String','w', ...
        'Style','text', ...
        'Tag','StaticText4');
b = uicontrol('Parent',H_STAB_OUT, ...
        'Units','points', ...
        'BackgroundColor',[0.752941 0.752941 0.752941], ...
        'FontSize',18, ...
        'HorizontalAlignment','left', ...
        'Position',[449.25 269.25 23.25 23.25], ...
        'String','q', ...
        'Style','text', ...
        'Tag','StaticText4');
b = uicontrol('Parent',H_STAB_OUT, ...
        'Units','points', ...
        'BackgroundColor',[0.752941 0.752941 0.752941], ...
        'FontSize',18, ...
        'HorizontalAlignment','left', ...
        'Position',[449.25 242.25 41.25 23.25], ...
        'String','theta', ...
        'Style','text', ...
        'Tag','StaticText4');
b = uicontrol('Parent',H_STAB_OUT, ...
        'Units','points', ...
        'FontSize',18, ...
        'HorizontalAlignment','left', ...
        'Position',[449.25 162 34.5 23.25], ...
        'String','phi', ...
        'Style','text', ...
        'Tag','StaticText4');
b = uicontrol('Parent',H_STAB_OUT, ...
        'Units','points', ...
        'FontSize',18, ...
        'HorizontalAlignment','left', ...
        'Position',[449.25 189 23.25 23.25], ...
        'String','p', ...
        'Style','text', ...
        'Tag','StaticText4');
b = uicontrol('Parent',H_STAB_OUT, ...
        'Units','points', ...
        'BackgroundColor',[0.752941 0.752941 0.752941], ...
        'FontSize',18, ...
        'HorizontalAlignment','left', ...
        'Position',[449.25 215.25 23.25 23.25], ...
        'String','v', ...
        'Style','text', ...
        'Tag','StaticText4');
b = uicontrol('Parent',H_STAB_OUT, ...
        'Units','points', ...
        'BackgroundColor',[0.752941 0.752941 0.752941], ...
        'FontSize',18, ...
        'HorizontalAlignment','left', ...
        'Position',[449.25 136.5 23.25 23.25], ...
        'String','r]', ...
        'Style','text', ...
        'Tag','StaticText4');
b = uicontrol('Parent',H_STAB_OUT, ...
```

```
        'Units','points', ...
        'BackgroundColor',[0.752941 0.752941 0.752941], ...
        'FontSize',18, ...
        'HorizontalAlignment','left', ...
        'Position',[508.5 243 50.25 23.25], ...
        'String','pedal]', ...
        'Style','text', ...
        'Tag','StaticText4');
b = uicontrol('Parent',H_STAB_OUT, ...
        'Units','points', ...
        'BackgroundColor',[0.752941 0.752941 0.752941], ...
        'FontSize',18, ...
        'HorizontalAlignment','left', ...
        'Position',[508.5 269.25 59.25 23.25], ...
        'String','lat. cyc.', ...
        'Style','text', ...
        'Tag','StaticText4');
b = uicontrol('Parent',H_STAB_OUT, ...
        'Units','points', ...
        'BackgroundColor',[0.752941 0.752941 0.752941], ...
        'FontSize',18, ...
        'HorizontalAlignment','left', ...
        'Position',[508.5 296.25 50.25 23.25], ...
        'String','col.', ...
        'Style','text', ...
        'Tag','StaticText4');
b = uicontrol('Parent',H_STAB_OUT, ...
        'Units','points', ...
        'BackgroundColor',[0.752941 0.752941 0.752941], ...
        'FontSize',18, ...
        'HorizontalAlignment','left', ...
        'Position',[449.25 322.5 23.25 23.25], ...
        'String','[u', ...
        'Style','text', ...
        'Tag','StaticText4');
b = uicontrol('Parent',H_STAB_OUT, ...
        'Units','points', ...
        'BackgroundColor',[0.752941 0.752941 0.752941], ...
        'FontSize',18, ...
        'HorizontalAlignment','left', ...
        'Position',[508.5 323.25 83.25 23.25], ...
        'String','[long. cyc.', ...
        'Style','text', ...
        'Tag','StaticText4');
```

25. stab_out_1_fcn.m

```
function stab_out_1_fcn(Action)

%     Switchyard Callback function for stab_out_1.m

%     Written for JANRAD version 6.0 by LT David A. Heathorn

global  S_STAB_INPUT_1 S_STAB_INPUT_2  S_PERF_INPUT S_PERF_OUTPUT A B
COUNT H_TF_RESP...
        H_STAB_OUT

if isempty(nargin)
   return
end

switch Action
case 'cr'
   time_freq_resp
   close(H_STAB_OUT)
   return

case 'save'
   sc_save
   return

case 'modin'
   COUNT=1;
   performance_input
   close(H_STAB_OUT)
   return


case 'print'
   set(gcf,'PaperOrientation','landscape')
   set(gcf,'PaperPosition',[.5 .5 10 7.5])
   print -dwinc

case 'return'
   janrad98
   close all

case 'quit'
   quit_gui

case 'about'
   about_janrad

case 'delta_input'
   performance_input


end
```

26. stability_and_control.m

```
function stability_and_control()

% JANRAD 98 VERSION 6.0

% This is the machine-generated representation of a Handle Graphics
object
% and its children.  Note that handle values may change when these
objects
% are re-created. This may cause problems with any callbacks written to
% depend on the value of the handle at the time the object was saved.
%
% To reopen this object, just type the name of the M-file at the MATLAB
% prompt. The M-file and its associated MAT-file must be on your path.

% Warning screen to indicate that Performance Module must be run before
% Stability and Control Analysis can be performed.

% Written for Janrad version 6.0 by LT David A. Heathorn

load stability_and_control

a = figure('Units','normalized', ...
      'Color',[0.8 0.8 0.8], ...
      'Colormap',mat0, ...
      'MenuBar','none', ...
      'Name','Stability and Control Error', ...
      'NumberTitle','off', ...
      'PointerShapeCData',mat1, ...
      'Position',[0.190625 0.383333 0.446875 0.34375], ...
   'Tag','Fig1');
b = uicontrol('Parent',a, ...
      'Units','normalized', ...
      'BackgroundColor',[0.752941 0.752941 0.752941], ...
      'Position',[0.013986 0.0363636 0.972028 0.933333], ...
      'Style','frame', ...
      'Tag','Frame1');


b = uicontrol('Parent',a, ...
      'Units','normalized', ...
      'Callback','close(gcf)', ...
      'FontSize',14, ...
      'FontWeight','bold', ...
      'Position',[0.388112 0.109091 0.202797 0.181818], ...
      'String','OK', ...
      'Tag','Pushbutton1');
b = uicontrol('Parent',a, ...
      'Units','normalized', ...
      'BackgroundColor',[0.752941 0.752941 0.752941], ...
      'FontSize',12, ...
      'FontWeight','bold', ...
      'Position',[0.0839161 0.515152 0.811189 0.345455], ...
   'String','Stability and Control can only be run after the Performance
Function', ...
      'Style','text', ...
```

```
        'Tag','StaticText1');
b = uicontrol('Parent',a, ...
        'Units','normalized', ...
        'BackgroundColor',[0.752941 0.752941 0.752941], ...
        'FontSize',12, ...
        'FontWeight','bold', ...
        'Position',[0.332168 0.357576 0.318182 0.127273], ...
        'String','SORRY!', ...
        'Style','text', ...
        'Tag','StaticText2');
```

27. stability_control_input_1.m

```
function stability_control_input_1()
% GUI window to display and/or edit input
% This is the machine-generated representation of a Handle Graphics
object
% and its children.  Note that handle values may change when these
objects
% are re-created. This may cause problems with any callbacks written to
% depend on the value of the handle at the time the object was saved.
%
% To reopen this object, just type the name of the M-file at the MATLAB
% prompt. The M-file and its associated MAT-file must be on your path.

% First GUI screen to input stability and control paramaters.

% Written for JANRAD version 6.0 by LT David A. Heathorn

load stability_control_input_1

global  H_STAB_IN1 S_STAB_INPUT_1 S_STAB_INPUT NAME S_SC_INPUT_1

if ~isempty(NAME)
       eval(['load ',NAME])
       unstructure_stab_input
       structure_stab_input_1
   else
      load create_new_stab
      structure_stab_input_1
   end

H_STAB_IN1 = figure('Units','normalized', ...
      'Color',[0.8 0.8 0.8], ...
      'Colormap',mat0, ...
      'CreateFcn','global MESH_VAL, MESH_VAL=0;;', ...
      'Name','Stability and Control Parameters page 1', ...
      'NumberTitle','off', ...
      'PointerShapeCData',mat1, ...
      'Position',[0.0025 0.0466667 0.9975 0.89], ...
   'Tag','Fig2');

b = uicontrol('Parent',H_STAB_IN1, ...
      'Units','normalized', ...
      'BackgroundColor',[0.752941 0.752941 0.752941], ...
      'Position',[0.0100251 0.513109 0.318296 0.385768], ...
      'Style','frame', ...
      'Tag','Frame1');
b = uicontrol('Parent',H_STAB_IN1, ...
      'Units','normalized', ...
      'BackgroundColor',[0.752941 0.752941 0.752941], ...
      'Position',[0.0100251 0.011236 0.317043 0.490637], ...
      'Style','frame', ...
      'Tag','Frame2');
b = uicontrol('Parent',H_STAB_IN1, ...
      'Units','normalized', ...
      'BackgroundColor',[0.752941 0.752941 0.752941], ...
      'Position',[0.338346 0.187266 0.322055 0.71161], ...
```

```
        'Style','frame', ...
        'Tag','Frame3');
b = uicontrol('Parent',H_STAB_IN1, ...
        'Units','normalized', ...
        'BackgroundColor',[0.752941 0.752941 0.752941], ...
        'Position',[0.679198 0.0674157 0.309524 0.835206], ...
        'Style','frame', ...
    'Tag','Frame4');

b = uimenu('Parent',H_STAB_IN1, ...
        'Label','JANRAD Options', ...
        'Tag','uimenu1');
c = uimenu('Parent',b, ...
        'Callback','performance_input_fcn quit', ...
        'Label','Quit JANRAD', ...
        'Tag','JANRAD OptionsSubuimenu1');
c = uimenu('Parent',b, ...
        'Callback','performance_input_fcn return', ...
        'Label','Return to Begining', ...
        'Tag','JANRAD OptionsSubuimenu1');
c = uimenu('Parent',b, ...
        'Callback','performance_input_fcn delta_input', ...
        'Enable','off', ...
        'Label','Change Input Parameters', ...
        'Tag','Subuimenu1');
c = uimenu('Parent',b, ...
        'Callback','performance_input_fcn about', ...
        'Label','About Janrad 98 ...', ...
        'Separator','on', ...
        'Tag','Subuimenu1');
b = uicontrol('Parent',H_STAB_IN1, ...
        'Units','normalized', ...
        'BackgroundColor',[0.752941 0.752941 0.752941], ...
        'FontSize',14, ...
        'FontWeight','bold', ...
        'Position',[0.122807 0.917603 0.763158 0.0692884], ...
        'String','STABILITY AND CONTROL PARAMETERS (PAGE 1 OF 2)', ...
        'Style','text', ...
        'Tag','StaticText3');
b = uicontrol('Parent',H_STAB_IN1, ...
        'Units','normalized', ...
        'BackgroundColor',[0.752941 0.752941 0.752941], ...
        'FontSize',14, ...
        'FontWeight','bold', ...
        'Position',[0.0175439 0.805243 0.294486 0.0898876], ...
        'String','MAIN ROTOR PARAMETERS', ...
        'Style','text', ...
        'Tag','StaticText1');
b = uicontrol('Parent',H_STAB_IN1, ...
        'Units','normalized', ...
        'BackgroundColor',[0.752941 0.752941 0.752941], ...
        'Position',[0.0197109 0.75 0.17477 0.0538462], ...
        'String','Flapping Moment of Inertia (slug-ft^2)', ...
        'Style','text', ...
        'Tag','StaticText2');
b = uicontrol('Parent',H_STAB_IN1, ...
        'Units','normalized', ...
```

```
      'BackgroundColor',[1 1 1], ...
      'Callback','Ib=get(gcbo,''String'');S_STAB_INPUT_1.Ib=str2num(Ib);
', ...
      'Position',[0.215506 0.751923 0.0985545 0.0480769], ...
      'String',Ib, ...
      'Style','edit', ...
      'Tag','EditText1');
b = uicontrol('Parent',H_STAB_IN1, ...
      'Units','normalized', ...
      'BackgroundColor',[0.752941 0.752941 0.752941], ...
      'Position',[0.0197109 0.694231 0.17477 0.0538462], ...
      'String','Hub Height Above Waterline (ft)', ...
      'Style','text', ...
      'Tag','StaticText2');
b = uicontrol('Parent',H_STAB_IN1, ...
      'Units','normalized', ...
      'BackgroundColor',[1 1 1], ...
      'Callback','hmd=get(gcbo,''String'');S_STAB_INPUT_1.hmd=str2num(hm
d);', ...
      'Position',[0.215506 0.694231 0.0985545 0.0480769], ...
      'String',hmd, ...
      'Style','edit', ...
      'Tag','EditText1');
b = uicontrol('Parent',H_STAB_IN1, ...
      'Units','normalized', ...
      'BackgroundColor',[0.752941 0.752941 0.752941], ...
      'Position',[0.021025 0.640385 0.17477 0.05], ...
      'String','Hub Fuselege Station (ft)', ...
      'Style','text', ...
      'Tag','StaticText2');
b = uicontrol('Parent',H_STAB_IN1, ...
      'Units','normalized', ...
      'BackgroundColor',[1 1 1], ...
      'Callback','lmd=get(gcbo,''String'');S_STAB_INPUT_1.lmd=str2num(lm
d);', ...
      'Position',[0.215506 0.640385 0.0985545 0.05], ...
      'String',lmd, ...
      'Style','edit', ...
      'Tag','EditText1');
b = uicontrol('Parent',H_STAB_IN1, ...
      'Units','normalized', ...
      'BackgroundColor',[0.752941 0.752941 0.752941], ...
      'Position',[0.0197109 0.582692 0.17477 0.05], ...
      'String','Hub Position Right of Buttline (ft)', ...
      'Style','text', ...
      'Tag','StaticText2');
b = uicontrol('Parent',H_STAB_IN1, ...
      'Units','normalized', ...
      'BackgroundColor',[1 1 1], ...
      'Callback','ymd=get(gcbo,''String'');S_STAB_INPUT_1.ymd=str2num(ym
d);', ...
      'Position',[0.21682 0.586538 0.0972405 0.0480769], ...
      'String',ymd, ...
      'Style','edit', ...
      'Tag','EditText1');
b = uicontrol('Parent',H_STAB_IN1, ...
      'Units','normalized', ...
```

```
        'BackgroundColor',[0.752941 0.752941 0.752941], ...
        'Position',[0.0197109 0.521154 0.17477 0.0538462], ...
        'String','Mast Incidence (negative fwd-degrees)', ...
        'Style','text', ...
        'Tag','StaticText2');
b = uicontrol('Parent',H_STAB_IN1, ...
        'Units','normalized', ...
        'BackgroundColor',[1 1 1], ...
        'Callback','im=get(gcbo,''String'');S_STAB_INPUT_1.im=str2num(im);
', ...
        'Position',[0.215506 0.528846 0.0985545 0.0480769], ...
        'String',im, ...
        'Style','edit', ...
        'Tag','EditText1');
b = uicontrol('Parent',H_STAB_IN1, ...
        'Units','normalized', ...
        'BackgroundColor',[0.752941 0.752941 0.752941], ...
        'FontSize',14, ...
        'FontWeight','bold', ...
        'Position',[0.022339 0.403846 0.296978 0.0903846], ...
        'String','VERTICAL FIN PARAMETERS', ...
        'Style','text', ...
        'Tag','StaticText1');
b = uicontrol('Parent',H_STAB_IN1, ...
        'Units','normalized', ...
        'BackgroundColor',[0.752941 0.752941 0.752941], ...
        'Position',[0.0315375 0.353846 0.173456 0.0480769], ...
        'String','Height Above waterline (ft)', ...
        'Style','text', ...
        'Tag','StaticText2');
b = uicontrol('Parent',H_STAB_IN1, ...
        'Units','normalized', ...
        'BackgroundColor',[1 1 1], ...
        'Callback','hvd=get(gcbo,''String'');S_STAB_INPUT_1.hvd=str2num(hv
d);', ...
        'Position',[0.215506 0.351923 0.0985545 0.0480769], ...
        'String',hvd, ...
        'Style','edit', ...
        'Tag','EditText1');
b = uicontrol('Parent',H_STAB_IN1, ...
        'Units','normalized', ...
        'BackgroundColor',[0.752941 0.752941 0.752941], ...
        'Position',[0.0341656 0.294231 0.17477 0.05], ...
        'String','Fuselage Station (ft)', ...
        'Style','text', ...
        'Tag','StaticText2');
b = uicontrol('Parent',H_STAB_IN1, ...
        'Units','normalized', ...
        'BackgroundColor',[1 1 1], ...
        'Callback','lvd=get(gcbo,''String'');S_STAB_INPUT_1.lvd=str2num(lv
d);', ...
        'Position',[0.215506 0.294231 0.0985545 0.05], ...
        'String',lvd, ...
        'Style','edit', ...
        'Tag','EditText1');
b = uicontrol('Parent',H_STAB_IN1, ...
        'Units','normalized', ...
```

```
        'BackgroundColor',[0.752941 0.752941 0.752941], ...
        'Position',[0.0341656 0.238462 0.173456 0.0480769], ...
        'String','Position Right of Buttline (ft)', ...
        'Style','text', ...
        'Tag','StaticText2');
b = uicontrol('Parent',H_STAB_IN1, ...
        'Units','normalized', ...
        'BackgroundColor',[1 1 1], ...
        'Callback','yvd=get(gcbo,''String'');S_STAB_INPUT_1.yvd=str2num(yv
d);', ...
        'Position',[0.215506 0.236538 0.0985545 0.0480769], ...
        'String',yvd, ...
        'Style','edit', ...
        'Tag','EditText1');
b = uicontrol('Parent',H_STAB_IN1, ...
        'Units','normalized', ...
        'BackgroundColor',[0.752941 0.752941 0.752941], ...
        'Position',[0.0341656 0.182692 0.173456 0.0480769], ...
        'String','Alpha Zero Lift (degrees)', ...
        'Style','text', ...
        'Tag','StaticText2');
b = uicontrol('Parent',H_STAB_IN1, ...
        'Units','normalized', ...
        'BackgroundColor',[1 1 1], ...
        'Callback','alplov=get(gcbo,''String'');S_STAB_INPUT_1.alplov=(str
2num(alplov));', ...
        'Position',[0.215506 0.180769 0.0985545 0.0480769], ...
        'String',alplov, ...
        'Style','edit', ...
        'Tag','EditText1');
b = uicontrol('Parent',H_STAB_IN1, ...
        'Units','normalized', ...
        'BackgroundColor',[0.752941 0.752941 0.752941], ...
        'Position',[0.0341656 0.126923 0.173456 0.0480769], ...
        'String','CL Max', ...
        'Style','text', ...
        'Tag','StaticText2');
b = uicontrol('Parent',H_STAB_IN1, ...
        'Units','normalized', ...
        'BackgroundColor',[1 1 1], ...
        'Callback','clvertmax=get(gcbo,''String'');S_STAB_INPUT_1.clvertma
x=str2num(clvertmax);', ...
        'Position',[0.215506 0.125 0.0985545 0.0480769], ...
        'String',clvertmax, ...
        'Style','edit', ...
        'Tag','EditText1');
b = uicontrol('Parent',H_STAB_IN1, ...
        'Units','normalized', ...
        'BackgroundColor',[0.752941 0.752941 0.752941], ...
        'Position',[0.0354796 0.0692308 0.173456 0.0557692], ...
        'String','Dynamic Pressure Ratio (page 489-Prouty)', ...
        'Style','text', ...
        'Tag','StaticText2');
b = uicontrol('Parent',H_STAB_IN1, ...
        'Units','normalized', ...
        'BackgroundColor',[1 1 1], ...
```

```
      'Callback','qvq=get(gcbo,''String'');S_STAB_INPUT_1.qvq=str2num(qv
q);', ...
      'Position',[0.215539 0.071161 0.0977444 0.0486891], ...
      'String',qvq, ...
      'Style','edit', ...
      'Tag','EditText1');
b = uicontrol('Parent',H_STAB_IN1, ...
      'Units','normalized', ...
      'BackgroundColor',[0.752941 0.752941 0.752941], ...
      'Position',[0.0350877 0.0262172 0.172932 0.0337079], ...
      'String','Lift Curve Slope', ...
      'Style','text', ...
      'Tag','StaticText2');
b = uicontrol('Parent',H_STAB_IN1, ...
      'Units','normalized', ...
      'BackgroundColor',[1 1 1], ...
      'Callback','av=get(gcbo,''String'');S_STAB_INPUT_1.av=str2num(av);
', ...
      'Position',[0.214286 0.0187266 0.0977444 0.0468165], ...
      'String',av, ...
      'Style','edit', ...
      'Tag','EditText1');
b = uicontrol('Parent',H_STAB_IN1, ...
      'Units','normalized', ...
      'BackgroundColor',[0.752941 0.752941 0.752941], ...
      'FontSize',14, ...
      'FontWeight','bold', ...
      'Position',[0.360419 0.796905 0.288336 0.0889749], ...
      'String','HORIZONTAL TAIL PARAMETERS', ...
      'Style','text', ...
      'Tag','StaticText1');
b = uicontrol('Parent',H_STAB_IN1, ...
      'Units','normalized', ...
      'BackgroundColor',[0.752941 0.752941 0.752941], ...
      'Position',[0.359109 0.727799 0.173001 0.046332], ...
      'String','Height Above Waterline (ft)', ...
      'Style','text', ...
      'Tag','StaticText2');
b = uicontrol('Parent',H_STAB_IN1, ...
      'Units','normalized', ...
      'BackgroundColor',[1 1 1], ...
      'Callback','hhd=get(gcbo,''String'');S_STAB_INPUT_1.hhd=str2num(hh
d);', ...
      'Position',[0.55308 0.729207 0.0982962 0.0483559], ...
      'String',hhd, ...
      'Style','edit', ...
      'Tag','EditText1');
b = uicontrol('Parent',H_STAB_IN1, ...
      'Units','normalized', ...
      'BackgroundColor',[0.752941 0.752941 0.752941], ...
      'Position',[0.360419 0.659574 0.174312 0.0483559], ...
      'String','Fuselage Station (ft)', ...
      'Style','text', ...
      'Tag','StaticText2');
b = uicontrol('Parent',H_STAB_IN1, ...
      'Units','normalized', ...
      'BackgroundColor',[1 1 1], ...
```

```
        'Callback','lhd=get(gcbo,''String'');S_STAB_INPUT_1.lhd=str2num(lh
d);', ...
        'Position',[0.554391 0.659574 0.0982962 0.0483559], ...
        'String',lhd, ...
        'Style','edit', ...
        'Tag','EditText1');
b = uicontrol('Parent',H_STAB_IN1, ...
        'Units','normalized', ...
        'BackgroundColor',[0.752941 0.752941 0.752941], ...
        'FontSize',6, ...
        'Position',[0.360053 0.590385 0.173456 0.0480769], ...
        'String','Position Right of Buttline (ft))', ...
        'Style','text', ...
        'Tag','StaticText2');
b = uicontrol('Parent',H_STAB_IN1, ...
        'Units','normalized', ...
        'BackgroundColor',[1 1 1], ...
        'Callback','yhd=get(gcbo,''String'');S_STAB_INPUT_1.yhd=str2num(yh
d);', ...
        'Position',[0.554391 0.59381 0.0982962 0.0483559], ...
        'String',yhd, ...
        'Style','edit', ...
        'Tag','EditText1');
b = uicontrol('Parent',H_STAB_IN1, ...
        'Units','normalized', ...
        'BackgroundColor',[0.752941 0.752941 0.752941], ...
        'Position',[0.360419 0.528046 0.174312 0.0483559], ...
        'String','Alpha Zero Lift (degrees)', ...
        'Style','text', ...
        'Tag','StaticText2');
b = uicontrol('Parent',H_STAB_IN1, ...
        'Units','normalized', ...
        'BackgroundColor',[1 1 1], ...
        'Callback','alploh=get(gcbo,''String'');S_STAB_INPUT_1.alploh=str2
num(alploh);', ...
        'Position',[0.554391 0.529981 0.0982962 0.0483559], ...
        'String',alploh, ...
        'Style','edit', ...
        'Tag','EditText1');
b = uicontrol('Parent',H_STAB_IN1, ...
        'Units','normalized', ...
        'BackgroundColor',[0.752941 0.752941 0.752941], ...
        'Position',[0.360419 0.464217 0.174312 0.0483559], ...
        'String','Angle of Incidence (degrees)', ...
        'Style','text', ...
        'Tag','StaticText2');
b = uicontrol('Parent',H_STAB_IN1, ...
        'Units','normalized', ...
        'BackgroundColor',[1 1 1], ...
        'Callback','ih=get(gcbo,''String'');S_STAB_INPUT_1.ih=str2num(ih);
', ...
        'Position',[0.554391 0.466151 0.0982962 0.0483559], ...
        'String',ih, ...
        'Style','edit', ...
        'Tag','EditText1');
b = uicontrol('Parent',H_STAB_IN1, ...
        'Units','normalized', ...
```

```
        'BackgroundColor',[0.752941 0.752941 0.752941], ...
        'Position',[0.360419 0.398453 0.174312 0.0483559], ...
        'String','Lift Curve Slope', ...
        'Style','text', ...
        'Tag','StaticText2');
b = uicontrol('Parent',H_STAB_IN1, ...
        'Units','normalized', ...
        'BackgroundColor',[1 1 1], ...
        'Callback','ah=get(gcbo,''String'');S_STAB_INPUT_1.ah=str2num(ah);
', ...
        'Position',[0.554391 0.400387 0.0982962 0.0483559], ...
        'String',ah, ...
        'Style','edit', ...
        'Tag','EditText1');
b = uicontrol('Parent',H_STAB_IN1, ...
        'Units','normalized', ...
        'BackgroundColor',[0.752941 0.752941 0.752941], ...
        'Position',[0.360053 0.328846 0.173456 0.0538462], ...
        'String','Dynamic Pressure Ratio (page 489 Prouty)', ...
        'Style','text', ...
        'Tag','StaticText2');
b = uicontrol('Parent',H_STAB_IN1, ...
        'Units','normalized', ...
        'BackgroundColor',[1 1 1], ...
        'Callback','qhq=get(gcbo,''String'');S_STAB_INPUT_1.qhq=str2num(qh
q);', ...
        'Position',[0.554391 0.336557 0.0982962 0.0483559], ...
        'String',qhq, ...
        'Style','edit', ...
        'Tag','EditText1');
b = uicontrol('Parent',H_STAB_IN1, ...
        'Units','normalized', ...
        'BackgroundColor',[0.752941 0.752941 0.752941], ...
        'Position',[0.358739 0.261538 0.17477 0.0538462], ...
        'String','Rotor Downwash Ratio (page 489 Prouty)', ...
        'Style','text', ...
        'Tag','StaticText2');
b = uicontrol('Parent',H_STAB_IN1, ...
        'Units','normalized', ...
        'BackgroundColor',[1 1 1], ...
        'Callback','vhv1=get(gcbo,''String'');S_STAB_INPUT_1.vhv1=str2num(
vhv1);', ...
        'Position',[0.554391 0.272727 0.0982962 0.0483559], ...
        'String',vhv1, ...
        'Style','edit', ...
        'Tag','EditText1');
b = uicontrol('Parent',H_STAB_IN1, ...
        'Units','normalized', ...
        'BackgroundColor',[0.752941 0.752941 0.752941], ...
        'Position',[0.360053 0.196154 0.173456 0.0557692], ...
        'String','Fuselage Downwash Ratio (page 489 Prouty)', ...
        'Style','text', ...
        'Tag','StaticText2');
b = uicontrol('Parent',H_STAB_IN1, ...
        'Units','normalized', ...
        'BackgroundColor',[1 1 1], ...
```

```
      'Callback','detafdalpfh=get(gcbo,''String'');S_STAB_INPUT_1.detafd
alpfh=str2num(detafdalpfh);', ...
      'Position',[0.554391 0.206963 0.0982962 0.0483559], ...
      'String',detafdalpfh, ...
      'Style','edit', ...
      'Tag','EditText1');
b = uicontrol('Parent',H_STAB_IN1, ...
      'Units','normalized', ...
      'BackgroundColor',[0.752941 0.752941 0.752941], ...
      'FontSize',14, ...
      'FontWeight','bold', ...
      'Position',[0.686763 0.798839 0.288336 0.0889749], ...
      'String','TAIL ROTOR PARAMETERS', ...
      'Style','text', ...
      'Tag','StaticText1');
b = uicontrol('Parent',H_STAB_IN1, ...
      'Units','normalized', ...
      'BackgroundColor',[0.752941 0.752941 0.752941], ...
      'Position',[0.688073 0.729207 0.173001 0.0464217], ...
      'String','Height Above Waterline (ft)', ...
      'Style','text', ...
      'Tag','StaticText2');
b = uicontrol('Parent',H_STAB_IN1, ...
      'Units','normalized', ...
      'BackgroundColor',[1 1 1], ...
      'Callback','htd=get(gcbo,''String'');S_STAB_INPUT_1.htd=str2num(ht
d);', ...
      'Position',[0.880734 0.729207 0.0982962 0.0483559], ...
      'String',htd, ...
      'Style','edit', ...
      'Tag','EditText1');
b = uicontrol('Parent',H_STAB_IN1, ...
      'Units','normalized', ...
      'BackgroundColor',[0.752941 0.752941 0.752941], ...
      'Position',[0.688073 0.659574 0.174312 0.0483559], ...
      'String','TR Fuselage Station (ft)', ...
      'Style','text', ...
      'Tag','StaticText2');
b = uicontrol('Parent',H_STAB_IN1, ...
      'Units','normalized', ...
      'BackgroundColor',[1 1 1], ...
      'Callback','ltd=get(gcbo,''String'');S_STAB_INPUT_1.ltd=str2num(lt
d);', ...
      'Position',[0.882045 0.659574 0.0982962 0.0483559], ...
      'String',ltd, ...
      'Style','edit', ...
      'Tag','EditText1');
b = uicontrol('Parent',H_STAB_IN1, ...
      'Units','normalized', ...
      'BackgroundColor',[0.752941 0.752941 0.752941], ...
      'FontSize',6, ...
      'Position',[0.688073 0.591876 0.174312 0.0483559], ...
      'String','Position Right of Buttline (ft)', ...
      'Style','text', ...
      'Tag','StaticText2');
b = uicontrol('Parent',H_STAB_IN1, ...
      'Units','normalized', ...
```

```
        'BackgroundColor',[1 1 1], ...
        'Callback','ytd=get(gcbo,''String'');S_STAB_INPUT_1.ytd=str2num(yt
d);', ...
        'Position',[0.882045 0.59381 0.0982962 0.0483559], ...
        'String',ytd, ...
        'Style','edit', ...
        'Tag','EditText1');
b = uicontrol('Parent',H_STAB_IN1, ...
        'Units','normalized', ...
        'BackgroundColor',[0.752941 0.752941 0.752941], ...
        'Position',[0.688073 0.528046 0.174312 0.0483559], ...
        'String','Number of Blades', ...
        'Style','text', ...
        'Tag','StaticText2');
b = uicontrol('Parent',H_STAB_IN1, ...
        'Units','normalized', ...
        'BackgroundColor',[1 1 1], ...
        'Callback','bt=get(gcbo,''String'');S_STAB_INPUT_1.bt=str2num(bt);
', ...
        'Position',[0.882045 0.529981 0.0982962 0.0483559], ...
        'String',bt, ...
        'Style','edit', ...
        'Tag','EditText1');
b = uicontrol('Parent',H_STAB_IN1, ...
        'Units','normalized', ...
        'BackgroundColor',[0.752941 0.752941 0.752941], ...
        'Position',[0.688073 0.464217 0.174312 0.0483559], ...
        'String','Blade Chord (ft)', ...
        'Style','text', ...
        'Tag','StaticText2');
b = uicontrol('Parent',H_STAB_IN1, ...
        'Units','normalized', ...
        'BackgroundColor',[1 1 1], ...
        'Callback','ct=get(gcbo,''String'');S_STAB_INPUT_1.ct=str2num(ct);
', ...
        'Position',[0.882045 0.466151 0.0982962 0.0483559], ...
        'String',ct, ...
        'Style','edit', ...
        'Tag','EditText1');
b = uicontrol('Parent',H_STAB_IN1, ...
        'Units','normalized', ...
        'BackgroundColor',[0.752941 0.752941 0.752941], ...
        'Position',[0.688073 0.398453 0.174312 0.0483559], ...
        'String','Blade Radius (ft)', ...
        'Style','text', ...
        'Tag','StaticText2');
b = uicontrol('Parent',H_STAB_IN1, ...
        'Units','normalized', ...
        'BackgroundColor',[1 1 1], ...
        'Callback','Rt=get(gcbo,''String'');S_STAB_INPUT_1.Rt=str2num(Rt);
', ...
        'Position',[0.882045 0.400387 0.0982962 0.0483559], ...
        'String',Rt, ...
        'Style','edit', ...
        'Tag','EditText1');
b = uicontrol('Parent',H_STAB_IN1, ...
        'Units','normalized', ...
```

```
        'BackgroundColor',[0.752941 0.752941 0.752941], ...
        'Position',[0.688073 0.334623 0.174312 0.0483559], ...
        'String','Lift Curve Slope', ...
        'Style','text', ...
        'Tag','StaticText2');
b = uicontrol('Parent',H_STAB_IN1, ...
        'Units','normalized', ...
        'BackgroundColor',[1 1 1], ...
        'Callback','at=get(gcbo,''String'');S_STAB_INPUT_1.at=str2num(at);
', ...
        'Position',[0.882045 0.336557 0.0982962 0.0483559], ...
        'String',at, ...
        'Style','edit', ...
        'Tag','EditText1');
b = uicontrol('Parent',H_STAB_IN1, ...
        'Units','normalized', ...
        'BackgroundColor',[0.752941 0.752941 0.752941], ...
        'Position',[0.687254 0.261538 0.17477 0.0538462], ...
        'String','Rotational Velocity (rad/sec)', ...
        'Style','text', ...
        'Tag','StaticText2');
b = uicontrol('Parent',H_STAB_IN1, ...
        'Units','normalized', ...
        'BackgroundColor',[1 1 1], ...
        'Callback','ohmt=get(gcbo,''String'');S_STAB_INPUT_1.ohmt=str2num(
ohmt);', ...
        'Position',[0.882045 0.272727 0.0982962 0.0483559], ...
        'String',ohmt, ...
        'Style','edit', ...
        'Tag','EditText1');
b = uicontrol('Parent',H_STAB_IN1, ...
        'Units','normalized', ...
        'BackgroundColor',[0.752941 0.752941 0.752941], ...
        'Position',[0.687254 0.192308 0.17477 0.0596154], ...
        'String','Flap Moment of Inertia (slug-ft^2)', ...
        'Style','text', ...
        'Tag','StaticText2');
b = uicontrol('Parent',H_STAB_IN1, ...
        'Units','normalized', ...
        'BackgroundColor',[1 1 1], ...
        'Callback','Ibt=get(gcbo,''String'');S_STAB_INPUT_1.Ibt=str2num(Ib
t);', ...
        'Position',[0.882045 0.206963 0.0982962 0.0483559], ...
        'String',Ibt, ...
        'Style','edit', ...
        'Tag','EditText1');
b = uicontrol('Parent',H_STAB_IN1, ...
        'Units','normalized', ...
        'BackgroundColor',[0.752941 0.752941 0.752941], ...
        'Position',[0.688073 0.141199 0.174312 0.0483559], ...
        'String','Delta-3 Angle (degrees)', ...
        'Style','text', ...
        'Tag','StaticText2');
b = uicontrol('Parent',H_STAB_IN1, ...
        'Units','normalized', ...
        'BackgroundColor',[1 1 1], ...
```

```
      'Callback','delta3=get(gcbo,''String'');S_STAB_INPUT_1.delta3=str2
num(delta3);', ...
      'Position',[0.882045 0.145068 0.0982962 0.0464217], ...
      'String',delta3, ...
      'Style','edit', ...
      'Tag','EditText1');
b = uicontrol('Parent',H_STAB_IN1, ...
      'Units','normalized', ...
      'BackgroundColor',[0.752941 0.752941 0.752941], ...
      'Position',[0.688073 0.0773694 0.174312 0.0483559], ...
      'String','Blade Twist (degrees)', ...
      'Style','text', ...
      'Tag','StaticText2');
b = uicontrol('Parent',H_STAB_IN1, ...
      'Units','normalized', ...
      'BackgroundColor',[1 1 1], ...
      'Callback','theta1t=get(gcbo,''String'');S_STAB_INPUT_1.theta1t=st
r2num(theta1t);', ...
      'Position',[0.882045 0.0793037 0.0982962 0.0483559], ...
      'String',theta1t, ...
      'Style','edit', ...
      'Tag','EditText1');
b = uicontrol('Parent',H_STAB_IN1, ...
      'Units','normalized', ...
   'Callback','global S_SC_INPUT_1;
S_SC_INPUT_1=S_STAB_INPUT_1;stab_control_input_fcn cont1', ...
      'FontSize',12, ...
      'FontWeight','bold', ...
      'Position',[0.51117 0.1 0.164258 0.0807692], ...
      'String','Continue >>', ...
      'Tag','Pushbutton4');
b = uicontrol('Parent',H_STAB_IN1, ...
      'Units','normalized', ...
      'Callback','stab_control_input_fcn back1', ...
      'FontSize',12, ...
      'FontWeight','bold', ...
      'Position',[0.339028 0.0115385 0.164258 0.0807692], ...
      'String','<< Back', ...
      'Tag','Pushbutton1');
b = uicontrol('Parent',H_STAB_IN1, ...
      'Units','normalized', ...
      'Callback','stab_control_input_fcn print', ...
      'FontSize',12, ...
      'FontWeight','bold', ...
      'Position',[0.509855 0.00961538 0.164258 0.0807692], ...
      'String','Print Screen', ...
      'Tag','Pushbutton2');
b = uicontrol('Parent',H_STAB_IN1, ...
      'Units','normalized', ...
      'Callback','stab_control_input_fcn cnx', ...
      'FontSize',12, ...
      'FontWeight','bold', ...
      'Position',[0.339028 0.1 0.164258 0.0807692], ...
      'String','Cancel', ...
      'Tag','Pushbutton3');

assignin('base','S_STAB_INPUT_1',S_STAB_INPUT_1);
```

```
assignin('base','S_SC_INPUT_1',S_SC_INPUT_1);
```

28. stability_control_input_2.m

```
function stability_control_input_2()
% This is the machine-generated representation of a Handle Graphics
object
% and its children.  Note that handle values may change when these
objects
% are re-created. This may cause problems with any callbacks written to
% depend on the value of the handle at the time the object was saved.
%
% To reopen this object, just type the name of the M-file at the MATLAB
% prompt. The M-file and its associated MAT-file must be on your path.

% Second GUI screen to input stability and control paramaters.

% Written for JANRAD version 6.0 by LT David A. Heathorn


load stability_control_input_2

global  H_STAB_IN1 H_STAB_IN2 S_STAB_INPUT_1 S_STAB_INPUT_2 S_STAB_INPUT
NAME S_SC_INPUT_2

if ~isempty(NAME)
   eval(['load ',NAME])
     unstructure_stab_input
     structure_stab_input_2
   else
    load create_new_stab_2
    structure_stab_input_2
   end

H_STAB_IN2 = figure('Units','normalized', ...
      'Color',[0.8 0.8 0.8], ...
      'Colormap',mat0, ...
      'CreateFcn','global MESH_VAL, MESH_VAL=0;;', ...
      'Name','Stability and Control Parameters page 2', ...
      'NumberTitle','off', ...
      'PointerShapeCData',mat1, ...
      'Position',[0.0025 0.0466667 0.9975 0.89], ...
   'Tag','Fig2');

b = uicontrol('Parent',H_STAB_IN2, ...
      'Units','normalized', ...
      'BackgroundColor',[0.752941 0.752941 0.752941], ...
      'Position',[0.011236 0.462777 0.313202 0.452716], ...
      'Style','frame', ...
      'Tag','Frame1');
b = uicontrol('Parent',H_STAB_IN2, ...
      'Units','normalized', ...
      'BackgroundColor',[0.752941 0.752941 0.752941], ...
      'Position',[0.0125313 0.00374532 0.313283 0.455056], ...
      'Style','frame', ...
      'Tag','Frame2');
b = uicontrol('Parent',H_STAB_IN2, ...
      'Units','normalized', ...
      'BackgroundColor',[0.752941 0.752941 0.752941], ...
```

```
        'Position',[0.344612 0.224719 0.315789 0.689139], ...
        'Style','frame', ...
        'Tag','Frame3');
b = uicontrol('Parent',H_STAB_IN2, ...
        'Units','normalized', ...
        'BackgroundColor',[0.752941 0.752941 0.752941], ...
        'Position',[0.682957 0.134831 0.310777 0.779026], ...
        'Style','frame', ...
        'Tag','Frame4');

b = uimenu('Parent',H_STAB_IN2, ...
        'Label','JANRAD Options', ...
        'Tag','uimenu1');
c = uimenu('Parent',b, ...
        'Callback','performance_input_fcn quit', ...
        'Label','Quit JANRAD', ...
        'Tag','JANRAD OptionsSubuimenu1');
c = uimenu('Parent',b, ...
        'Callback','performance_input_fcn return', ...
        'Label','Return to Begining', ...
        'Tag','JANRAD OptionsSubuimenu1');
c = uimenu('Parent',b, ...
        'Callback','performance_input_fcn delta_input', ...
        'Enable','off', ...
        'Label','Change Input Parameters', ...
        'Tag','Subuimenu1');
c = uimenu('Parent',b, ...
        'Callback','performance_input_fcn about', ...
        'Label','About Janrad 98 ...', ...
        'Separator','on', ...
        'Tag','Subuimenu1');
b = uicontrol('Parent',H_STAB_IN2, ...
        'Units','normalized', ...
        'BackgroundColor',[0.752941 0.752941 0.752941], ...
        'FontSize',14, ...
        'FontWeight','bold', ...
        'Position',[0.106516 0.917603 0.763158 0.0674157], ...
        'String','STABILITY AND CONTROL PARAMETERS (PAGE 2 OF 2)', ...
        'Style','text', ...
        'Tag','StaticText3');
b = uicontrol('Parent',H_STAB_IN2, ...
        'Units','normalized', ...
        'BackgroundColor',[0.752941 0.752941 0.752941], ...
        'FontSize',14, ...
        'FontWeight','bold', ...
        'Position',[0.0225564 0.816479 0.290727 0.0898876], ...
        'String','RIGGING PARAMETERS', ...
        'Style','text', ...
        'Tag','StaticText1');
b = uicontrol('Parent',H_STAB_IN2, ...
        'Units','normalized', ...
        'BackgroundColor',[0.752941 0.752941 0.752941], ...
        'Position',[0.0160214 0.763359 0.174667 0.0534351], ...
        'String','Long Cyclic Pitch per inch deflection (degrees/in)', ...
        'Style','text', ...
        'Tag','StaticText2');
b = uicontrol('Parent',H_STAB_IN2, ...
```

```
      'Units','normalized', ...
      'BackgroundColor',[1 1 1], ...
      'Callback','db1mddele=get(gcbo,''String'');S_STAB_INPUT_2.db1mddel
e=str2num(db1mddele);', ...
      'Position',[0.217867 0.76673 0.0985545 0.0478011], ...
      'String',db1mddele, ...
      'Style','edit', ...
      'Tag','EditText1');
b = uicontrol('Parent',H_STAB_IN2, ...
      'Units','normalized', ...
      'BackgroundColor',[0.752941 0.752941 0.752941], ...
      'Position',[0.0160214 0.708015 0.1749 0.0534351], ...
      'String','Lateral Cyclic Pitch per inch deflection (deg/in)', ...
      'Style','text', ...
      'Tag','StaticText2');
b = uicontrol('Parent',H_STAB_IN2, ...
      'Units','normalized', ...
      'BackgroundColor',[1 1 1], ...
      'Callback','da1mddela=get(gcbo,''String'');S_STAB_INPUT_2.da1mddel
a=str2num(da1mddela);', ...
      'Position',[0.217867 0.709369 0.0985545 0.0478011], ...
      'String',da1mddela, ...
      'Style','edit', ...
      'Tag','EditText1');
b = uicontrol('Parent',H_STAB_IN2, ...
      'Units','normalized', ...
      'BackgroundColor',[0.752941 0.752941 0.752941], ...
      'Position',[0.0160214 0.650763 0.173565 0.0515267], ...
      'String','Collective pitch per inch deflection (deg/in)', ...
      'Style','text', ...
      'Tag','StaticText2');
b = uicontrol('Parent',H_STAB_IN2, ...
      'Units','normalized', ...
      'BackgroundColor',[1 1 1], ...
      'Callback','dthetomddelc=get(gcbo,''String'');S_STAB_INPUT_2.dthet
omddelc=str2num(dthetomddelc);', ...
      'Position',[0.217867 0.652008 0.0985545 0.0516252], ...
      'String',dthetomddelc, ...
      'Style','edit', ...
      'Tag','EditText1');
b = uicontrol('Parent',H_STAB_IN2, ...
      'Units','normalized', ...
      'BackgroundColor',[0.752941 0.752941 0.752941], ...
      'Position',[0.0160214 0.590566 0.1749 0.0528302], ...
      'String','theta0t/pedal deflection (deg/in or deg/deg)', ...
      'Style','text', ...
      'Tag','StaticText2');
b = uicontrol('Parent',H_STAB_IN2, ...
      'Units','normalized', ...
      'BackgroundColor',[1 1 1], ...
      'Callback','dthetotddelp=get(gcbo,''String'');S_STAB_INPUT_2.dthet
otddelp=str2num(dthetotddelp);', ...
      'Position',[0.217867 0.594646 0.0985545 0.0516252], ...
      'String',dthetotddelp, ...
      'Style','edit', ...
      'Tag','EditText1');
b = uicontrol('Parent',H_STAB_IN2, ...
```

```
      'Units','normalized', ...
      'BackgroundColor',[0.752941 0.752941 0.752941], ...
      'Position',[0.0160214 0.530189 0.1749 0.0528302], ...
      'String','NOTAR slv twst/defl  (deg. or in. travel) 1000 for TR',
...
      'Style','text', ...
      'Tag','StaticText2');
b = uicontrol('Parent',H_STAB_IN2, ...
      'Units','normalized', ...
      'BackgroundColor',[1 1 1], ...
      'Callback','sidearm=get(gcbo,''String'');S_STAB_INPUT_2.sidearm=st
r2num(sidearm);', ...
      'Position',[0.218045 0.535581 0.0977444 0.0468165], ...
      'String',sidearm, ...
      'Style','edit', ...
      'Tag','EditText1');
b = uicontrol('Parent',H_STAB_IN2, ...
      'Units','normalized', ...
      'BackgroundColor',[0.752941 0.752941 0.752941], ...
      'Position',[0.0160214 0.466038 0.1749 0.0509434], ...
      'String','Max Rudder Deflection (deg or in. travel)', ...
      'Style','text', ...
      'Tag','StaticText2');
b = uicontrol('Parent',H_STAB_IN2, ...
      'Units','normalized', ...
      'BackgroundColor',[1 1 1], ...
      'Callback','maxr=get(gcbo,''String'');S_STAB_INPUT_2.maxr=str2num(
maxr);', ...
      'Position',[0.217623 0.479245 0.0987984 0.0471698], ...
      'String',maxr, ...
      'Style','edit', ...
      'Tag','EditText1');
b = uicontrol('Parent',H_STAB_IN2, ...
      'Units','normalized', ...
      'BackgroundColor',[0.752941 0.752941 0.752941], ...
      'FontSize',14, ...
      'FontWeight','bold', ...
      'Position',[0.0200501 0.400749 0.298246 0.0505618], ...
      'String','NOTAR PARAMETERS', ...
      'Style','text', ...
      'Tag','StaticText1');
b = uicontrol('Parent',H_STAB_IN2, ...
      'Units','normalized', ...
      'BackgroundColor',[0.752941 0.752941 0.752941], ...
      'Position',[0.0291139 0.357414 0.173418 0.0361217], ...
      'String','Height Above waterline (ft)', ...
      'Style','text', ...
      'Tag','StaticText2');
b = uicontrol('Parent',H_STAB_IN2, ...
      'Units','normalized', ...
      'BackgroundColor',[1 1 1], ...
      'Callback','htnd=get(gcbo,''String'');S_STAB_INPUT_2.htnd=str2num(
htnd);', ...
      'Position',[0.21682 0.351923 0.0985545 0.0480769], ...
      'String',htnd, ...
      'Style','edit', ...
      'Tag','EditText1');
```

```
b = uicontrol('Parent',H_STAB_IN2, ...
      'Units','normalized', ...
      'BackgroundColor',[0.752941 0.752941 0.752941], ...
      'Position',[0.0293725 0.290566 0.1749 0.0509434], ...
      'String','Boom Fuselage Station (ft)', ...
      'Style','text', ...
      'Tag','StaticText2');
b = uicontrol('Parent',H_STAB_IN2, ...
      'Units','normalized', ...
      'BackgroundColor',[1 1 1], ...
      'Callback','ltnd=get(gcbo,''String'');S_STAB_INPUT_2.ltnd=str2num(
ltnd);', ...
      'Position',[0.21682 0.294231 0.0985545 0.05], ...
      'String',ltnd, ...
      'Style','edit', ...
      'Tag','EditText1');
b = uicontrol('Parent',H_STAB_IN2, ...
      'Units','normalized', ...
      'BackgroundColor',[0.752941 0.752941 0.752941], ...
      'Position',[0.0293725 0.233962 0.173565 0.0490566], ...
      'String','Boom Position Right of Buttline  (ft)', ...
      'Style','text', ...
      'Tag','StaticText2');
b = uicontrol('Parent',H_STAB_IN2, ...
      'Units','normalized', ...
      'BackgroundColor',[1 1 1], ...
      'Callback','ytnd=get(gcbo,''String'');S_STAB_INPUT_2.ytnd=str2num(
ytnd);', ...
      'Position',[0.21682 0.236538 0.0985545 0.0480769], ...
      'String',ytnd, ...
      'Style','edit', ...
      'Tag','EditText1');
b = uicontrol('Parent',H_STAB_IN2, ...
      'Units','normalized', ...
      'BackgroundColor',[0.752941 0.752941 0.752941], ...
      'Position',[0.0293725 0.19434 0.173565 0.0358491], ...
      'String','NOTAR diameter (ft))', ...
      'Style','text', ...
      'Tag','StaticText2');
b = uicontrol('Parent',H_STAB_IN2, ...
      'Units','normalized', ...
      'BackgroundColor',[1 1 1], ...
      'Callback','dian=get(gcbo,''String'');S_STAB_INPUT_2.dian=str2num(
dian);', ...
      'Position',[0.21682 0.180769 0.0985545 0.0480769], ...
      'String',dian, ...
      'Style','edit', ...
      'Tag','EditText1');
b = uicontrol('Parent',H_STAB_IN2, ...
      'Units','normalized', ...
      'BackgroundColor',[0.752941 0.752941 0.752941], ...
      'Position',[0.0293725 0.120755 0.173565 0.0584906], ...
      'String','Swirl Angle at Boom (degrees)', ...
      'Style','text', ...
      'Tag','StaticText2');
b = uicontrol('Parent',H_STAB_IN2, ...
      'Units','normalized', ...
```

```
        'BackgroundColor',[1 1 1], ...
        'Callback','swirl=get(gcbo,''String'');S_STAB_INPUT_2.swirl=str2nu
m(swirl);', ...
        'Position',[0.21682 0.125 0.0985545 0.0480769], ...
        'String',swirl, ...
        'Style','edit', ...
        'Tag','EditText1');
b = uicontrol('Parent',H_STAB_IN2, ...
        'Units','normalized', ...
        'BackgroundColor',[0.752941 0.752941 0.752941], ...
        'Position',[0.0293725 0.0792453 0.173565 0.0339623], ...
        'String','NOTAR Max Force (lbs)', ...
        'Style','text', ...
        'Tag','StaticText2');
b = uicontrol('Parent',H_STAB_IN2, ...
        'Units','normalized', ...
        'BackgroundColor',[1 1 1], ...
        'Callback','Ytmaxn=get(gcbo,''String'');S_STAB_INPUT_2.Ytmaxn=str2
num(Ytmaxn);', ...
        'Position',[0.215506 0.0711538 0.0998686 0.0480769], ...
        'String',Ytmaxn, ...
        'Style','edit', ...
        'Tag','EditText1');
b = uicontrol('Parent',H_STAB_IN2, ...
        'Units','normalized', ...
        'BackgroundColor',[0.752941 0.752941 0.752941], ...
        'Position',[0.0293725 0.0113208 0.173565 0.0528302], ...
        'String','Thruster Fuselage Station (ft)', ...
        'Style','text', ...
        'Tag','StaticText2');
b = uicontrol('Parent',H_STAB_IN2, ...
        'Units','normalized', ...
        'BackgroundColor',[1 1 1], ...
        'Callback','lttnd=get(gcbo,''String'');S_STAB_INPUT_2.lttnd=str2nu
m(lttnd);', ...
        'Position',[0.21682 0.0173077 0.0985545 0.0480769], ...
        'String',lttnd, ...
        'Style','edit', ...
        'Tag','EditText1');
b = uicontrol('Parent',H_STAB_IN2, ...
        'Units','normalized', ...
        'BackgroundColor',[0.752941 0.752941 0.752941], ...
        'FontSize',14, ...
        'FontWeight','bold', ...
        'Position',[0.358396 0.754682 0.286967 0.151685], ...
        'String','CG LOCATION & INERTIAS/FUSELAGE PARAMETERS', ...
        'Style','text', ...
        'Tag','StaticText1');
b = uicontrol('Parent',H_STAB_IN2, ...
        'Units','normalized', ...
        'BackgroundColor',[0.752941 0.752941 0.752941], ...
        'Position',[0.356475 0.704198 0.171278 0.0458015], ...
        'String','CG Height Above Waterline  (ft)', ...
        'Style','text', ...
        'Tag','StaticText2');
b = uicontrol('Parent',H_STAB_IN2, ...
        'Units','normalized', ...
```

```
      'BackgroundColor',[1 1 1], ...
      'Callback','zcg=get(gcbo,''String'');S_STAB_INPUT_2.zcg=str2num(zc
g);', ...
      'Position',[0.550591 0.705545 0.0985545 0.0478011], ...
      'String',zcg, ...
      'Style','edit', ...
      'Tag','EditText1');
b = uicontrol('Parent',H_STAB_IN2, ...
      'Units','normalized', ...
      'BackgroundColor',[0.752941 0.752941 0.752941], ...
      'Position',[0.356475 0.633588 0.172596 0.0496183], ...
      'String','CG Fuselage Station (ft)', ...
      'Style','text', ...
      'Tag','StaticText2');
b = uicontrol('Parent',H_STAB_IN2, ...
      'Units','normalized', ...
      'BackgroundColor',[1 1 1], ...
      'Callback','xcg=get(gcbo,''String'');S_STAB_INPUT_2.xcg=str2num(xc
g);', ...
      'Position',[0.551905 0.634799 0.0985545 0.0497132], ...
      'String',xcg, ...
      'Style','edit', ...
      'Tag','EditText1');
b = uicontrol('Parent',H_STAB_IN2, ...
      'Units','normalized', ...
      'BackgroundColor',[0.752941 0.752941 0.752941], ...
      'FontSize',6, ...
      'Position',[0.356475 0.564885 0.172596 0.0477099], ...
      'String','CG Position Right of Buttline (ft))', ...
      'Style','text', ...
      'Tag','StaticText2');
b = uicontrol('Parent',H_STAB_IN2, ...
      'Units','normalized', ...
      'BackgroundColor',[1 1 1], ...
      'Callback','ycg=get(gcbo,''String'');S_STAB_INPUT_2.ycg=str2num(yc
g);', ...
      'Position',[0.551905 0.56979 0.0985545 0.0478011], ...
      'String',ycg, ...
      'Style','edit', ...
      'Tag','EditText1');
b = uicontrol('Parent',H_STAB_IN2, ...
      'Units','normalized', ...
      'BackgroundColor',[0.752941 0.752941 0.752941], ...
      'Position',[0.356475 0.50478 0.173456 0.0478011], ...
      'String','Ixx (slug ft^2)', ...
      'Style','text', ...
      'Tag','StaticText2');
b = uicontrol('Parent',H_STAB_IN2, ...
      'Units','normalized', ...
      'BackgroundColor',[1 1 1], ...
      'Callback','Ixx=get(gcbo,''String'');S_STAB_INPUT_2.Ixx=str2num(Ix
x);', ...
      'Position',[0.551905 0.506692 0.0985545 0.0478011], ...
      'String',Ixx, ...
      'Style','edit', ...
      'Tag','EditText1');
b = uicontrol('Parent',H_STAB_IN2, ...
```

```
        'Units','normalized', ...
        'BackgroundColor',[0.752941 0.752941 0.752941], ...
        'Position',[0.356475 0.438931 0.173565 0.0496183], ...
        'String','Iyy (slug ft^2)', ...
        'Style','text', ...
        'Tag','StaticText2');
b = uicontrol('Parent',H_STAB_IN2, ...
        'Units','normalized', ...
        'BackgroundColor',[1 1 1], ...
        'Callback','Iyy=get(gcbo,''String'');S_STAB_INPUT_2.Iyy=str2num(Iy
y);', ...
        'Position',[0.551905 0.441683 0.0985545 0.0497132], ...
        'String',Iyy, ...
        'Style','edit', ...
        'Tag','EditText1');
b = uicontrol('Parent',H_STAB_IN2, ...
        'Units','normalized', ...
        'BackgroundColor',[0.752941 0.752941 0.752941], ...
        'Position',[0.356475 0.374761 0.173456 0.0478011], ...
        'String','Izz (slug ft^2)', ...
        'Style','text', ...
        'Tag','StaticText2');
b = uicontrol('Parent',H_STAB_IN2, ...
        'Units','normalized', ...
        'BackgroundColor',[1 1 1], ...
        'Callback','Izz=get(gcbo,''String'');S_STAB_INPUT_2.Izz=str2num(Iz
z);', ...
        'Position',[0.551905 0.376673 0.0985545 0.0478011], ...
        'String',Izz, ...
        'Style','edit', ...
        'Tag','EditText1');
b = uicontrol('Parent',H_STAB_IN2, ...
        'Units','normalized', ...
        'BackgroundColor',[0.752941 0.752941 0.752941], ...
        'Position',[0.356475 0.304015 0.173456 0.0554493], ...
        'String','Ixz (slug ft^2)', ...
        'Style','text', ...
        'Tag','StaticText2');
b = uicontrol('Parent',H_STAB_IN2, ...
        'Units','normalized', ...
        'BackgroundColor',[1 1 1], ...
        'Callback','Ixz=get(gcbo,''String'');S_STAB_INPUT_2.Ixz=str2num(Ix
z);', ...
        'Position',[0.551905 0.313576 0.0985545 0.0478011], ...
        'String',Ixz, ...
        'Style','edit', ...
        'Tag','EditText1');
b = uicontrol('Parent',H_STAB_IN2, ...
        'Units','normalized', ...
        'BackgroundColor',[0.752941 0.752941 0.752941], ...
        'Position',[0.356475 0.237094 0.17477 0.0535373], ...
        'String','Fuselage Downwash Ratio (page 513 Prouty)', ...
        'Style','text', ...
        'Tag','StaticText2');
b = uicontrol('Parent',H_STAB_IN2, ...
        'Units','normalized', ...
        'BackgroundColor',[1 1 1], ...
```

```
    'Callback','vfv1=get(gcbo,''String'');S_STAB_INPUT_2.vfv1=str2num(
vfv1);', ...
    'Position',[0.551905 0.248566 0.0985545 0.0478011], ...
    'String',vfv1, ...
    'Style','edit', ...
    'Tag','EditText1');
b = uicontrol('Parent',H_STAB_IN2, ...
    'Units','normalized', ...
    'BackgroundColor',[0.752941 0.752941 0.752941], ...
    'FontSize',14, ...
    'FontWeight','bold', ...
    'Position',[0.68797 0.797753 0.289474 0.0898876], ...
    'String','WING PARAMETERS', ...
    'Style','text', ...
    'Tag','StaticText1');
b = uicontrol('Parent',H_STAB_IN2, ...
    'Units','normalized', ...
    'BackgroundColor',[0.752941 0.752941 0.752941], ...
    'Position',[0.68594 0.745698 0.173456 0.0458891], ...
    'String','Height Above Waterline (ft)', ...
    'Style','text', ...
    'Tag','StaticText2');
b = uicontrol('Parent',H_STAB_IN2, ...
    'Units','normalized', ...
    'BackgroundColor',[1 1 1], ...
    'Callback','hwd=get(gcbo,''String'');S_STAB_INPUT_2.hwd=str2num(hw
d);', ...
    'Position',[0.88042 0.74761 0.0985545 0.0478011], ...
    'String',hwd, ...
    'Style','edit', ...
    'Tag','EditText1');
b = uicontrol('Parent',H_STAB_IN2, ...
    'Units','normalized', ...
    'BackgroundColor',[0.752941 0.752941 0.752941], ...
    'Position',[0.68594 0.674952 0.17477 0.0497132], ...
    'String','Fuselage Station (ft)', ...
    'Style','text', ...
    'Tag','StaticText2');
b = uicontrol('Parent',H_STAB_IN2, ...
    'Units','normalized', ...
    'BackgroundColor',[1 1 1], ...
    'Callback','lwd=get(gcbo,''String'');S_STAB_INPUT_2.lwd=str2num(lw
d);', ...
    'Position',[0.881735 0.676864 0.0985545 0.0497132], ...
    'String',lwd, ...
    'Style','edit', ...
    'Tag','EditText1');
b = uicontrol('Parent',H_STAB_IN2, ...
    'Units','normalized', ...
    'BackgroundColor',[0.752941 0.752941 0.752941], ...
    'FontSize',6, ...
    'Position',[0.68594 0.609943 0.173456 0.0478011], ...
    'String','Position Right of Buttline (ft)', ...
    'Style','text', ...
    'Tag','StaticText2');
b = uicontrol('Parent',H_STAB_IN2, ...
    'Units','normalized', ...
```

```
        'BackgroundColor',[1 1 1], ...
        'Callback','ywd=get(gcbo,''String'');S_STAB_INPUT_2.ywd=str2num(yw
d);', ...
        'Position',[0.881735 0.611855 0.0985545 0.0478011], ...
        'String',ywd, ...
        'Style','edit', ...
        'Tag','EditText1');
b = uicontrol('Parent',H_STAB_IN2, ...
        'Units','normalized', ...
        'BackgroundColor',[0.752941 0.752941 0.752941], ...
        'Position',[0.68594 0.544933 0.17477 0.0478011], ...
        'String','Alpha Zero Lift (degrees)', ...
        'Style','text', ...
        'Tag','StaticText2');
b = uicontrol('Parent',H_STAB_IN2, ...
        'Units','normalized', ...
        'BackgroundColor',[1 1 1], ...
        'Callback','alplow=get(gcbo,''String'');S_STAB_INPUT_2.alplow=str2
num(alplow);', ...
        'Position',[0.881735 0.548757 0.0985545 0.0478011], ...
        'String',alplow, ...
        'Style','edit', ...
        'Tag','EditText1');
b = uicontrol('Parent',H_STAB_IN2, ...
        'Units','normalized', ...
        'BackgroundColor',[0.752941 0.752941 0.752941], ...
        'Position',[0.68594 0.481836 0.173456 0.0497132], ...
        'String','Angle of Incidence (degrees)', ...
        'Style','text', ...
        'Tag','StaticText2');
b = uicontrol('Parent',H_STAB_IN2, ...
        'Units','normalized', ...
        'BackgroundColor',[1 1 1], ...
        'Callback','iw=get(gcbo,''String'');S_STAB_INPUT_2.iw=str2num(iw);
', ...
        'Position',[0.881735 0.483748 0.0985545 0.0497132], ...
        'String',iw, ...
        'Style','edit', ...
        'Tag','EditText1');
b = uicontrol('Parent',H_STAB_IN2, ...
        'Units','normalized', ...
        'BackgroundColor',[0.752941 0.752941 0.752941], ...
        'Position',[0.68594 0.355641 0.17477 0.0478011], ...
        'String','Tip Chord (ft)', ...
        'Style','text', ...
        'Tag','StaticText2');
b = uicontrol('Parent',H_STAB_IN2, ...
        'Units','normalized', ...
        'BackgroundColor',[1 1 1], ...
        'Callback','ctw=get(gcbo,''String'');S_STAB_INPUT_2.ctw=str2num(ct
w);', ...
        'Position',[0.881735 0.418738 0.0985545 0.0478011], ...
        'String',ctw, ...
        'Style','edit', ...
        'Tag','EditText1');
b = uicontrol('Parent',H_STAB_IN2, ...
        'Units','normalized', ...
```

```
        'BackgroundColor',[0.752941 0.752941 0.752941], ...
        'Position',[0.68594 0.416826 0.17477 0.0478011], ...
        'String','Lift Curve Slope', ...
        'Style','text', ...
        'Tag','StaticText2');
b = uicontrol('Parent',H_STAB_IN2, ...
        'Units','normalized', ...
        'BackgroundColor',[1 1 1], ...
        'Callback','aw=get(gcbo,''String'');S_STAB_INPUT_2.aw=str2num(aw);
', ...
        'Position',[0.881735 0.355641 0.0985545 0.0478011], ...
        'String',aw, ...
        'Style','edit', ...
        'Tag','EditText1');
b = uicontrol('Parent',H_STAB_IN2, ...
        'Units','normalized', ...
        'BackgroundColor',[0.752941 0.752941 0.752941], ...
        'Position',[0.68594 0.286807 0.17477 0.0535373], ...
        'String','Root Chord (ft)', ...
        'Style','text', ...
        'Tag','StaticText2');
b = uicontrol('Parent',H_STAB_IN2, ...
        'Units','normalized', ...
        'BackgroundColor',[1 1 1], ...
        'Callback','crw=get(gcbo,''String'');S_STAB_INPUT_2.crw=str2num(cr
w);', ...
        'Position',[0.881735 0.290631 0.0985545 0.0478011], ...
        'String',crw, ...
        'Style','edit', ...
        'Tag','EditText1');
b = uicontrol('Parent',H_STAB_IN2, ...
        'Units','normalized', ...
        'BackgroundColor',[0.752941 0.752941 0.752941], ...
        'Position',[0.68594 0.217973 0.17477 0.0592734], ...
        'String','Rotor Downwash Ratio (page 489-Prouty)', ...
        'Style','text', ...
        'Tag','StaticText2');
b = uicontrol('Parent',H_STAB_IN2, ...
        'Units','normalized', ...
        'BackgroundColor',[1 1 1], ...
        'Callback','vwv1=get(gcbo,''String'');S_STAB_INPUT_2.vwv1=str2num(
vwv1);', ...
        'Position',[0.881735 0.225621 0.0985545 0.0478011], ...
        'String',vwv1, ...
        'Style','edit', ...
        'Tag','EditText1');
b = uicontrol('Parent',H_STAB_IN2, ...
        'Units','normalized', ...
        'BackgroundColor',[0.752941 0.752941 0.752941], ...
        'Position',[0.68594 0.1587 0.173456 0.0497132], ...
        'String','Fuselage Downwash Ratio (page 489-Prouty)', ...
        'Style','text', ...
        'Tag','StaticText2');
b = uicontrol('Parent',H_STAB_IN2, ...
        'Units','normalized', ...
        'BackgroundColor',[1 1 1], ...
```

```
      'Callback','detafdalpfw=get(gcbo,''String'');S_STAB_INPUT_2.detafd
alpfw=str2num(detafdalpfw);', ...
      'Position',[0.881735 0.162524 0.0985545 0.0478011], ...
      'String',detafdalpfw, ...
      'Style','edit', ...
      'Tag','EditText1');
b = uicontrol('Parent',H_STAB_IN2, ...
      'Units','normalized', ...
      'Callback','global S_SC_INPUT_2;
S_SC_INPUT_2=S_STAB_INPUT_2;stab_control_input_fcn cont2', ...
      'FontSize',12, ...
      'FontWeight','bold', ...
      'Position',[0.51117 0.1 0.164258 0.0807692], ...
      'String','Continue >>', ...
      'Tag','Pushbutton4');
b = uicontrol('Parent',H_STAB_IN2, ...
      'Units','normalized', ...
      'Callback','stab_control_input_fcn back2', ...
      'FontSize',12, ...
      'FontWeight','bold', ...
      'Position',[0.339028 0.0115385 0.164258 0.0807692], ...
      'String','<< Back', ...
      'Tag','Pushbutton1');
b = uicontrol('Parent',H_STAB_IN2, ...
      'Units','normalized', ...
      'Callback','stab_control_input_fcn print', ...
      'FontSize',12, ...
      'FontWeight','bold', ...
      'Position',[0.510013 0.0114504 0.164219 0.0801527], ...
      'String','Print Screen', ...
      'Tag','Pushbutton2');
b = uicontrol('Parent',H_STAB_IN2, ...
      'Units','normalized', ...
      'Callback','stab_control_input_fcn cnx', ...
      'FontSize',12, ...
      'FontWeight','bold', ...
      'Position',[0.339119 0.101145 0.164219 0.0801527], ...
      'String','Cancel', ...
      'Tag','Pushbutton3');
b = uicontrol('Parent',H_STAB_IN2, ...
      'Units','normalized', ...
      'BackgroundColor',[0.752941 0.752941 0.752941], ...
      'FontSize',14, ...
      'FontWeight','bold', ...
      'Position',[0.755357 1.09524 0.296199 0.0831721], ...
      'String','MAIN ROTOR PARAMETERS', ...
      'Style','text', ...
      'Tag','StaticText1');
b = uicontrol('Parent',H_STAB_IN2, ...
      'Units','normalized', ...
      'BackgroundColor',[0.752941 0.752941 0.752941], ...
      'Position',[0.0160214 0.760377 0.1749 0.0528302], ...
      'String','Long Cyclic Pitch per inch deflection (degrees/in)', ...
      'Style','text', ...
      'Tag','StaticText2');

assignin('base','S_STAB_INPUT_2',S_STAB_INPUT_2);
```

```
assignin('base','S_SC_INPUT_2',S_SC_INPUT_2);
```

29. stby_scrn.m

```
function stby_scrn()
% This is the machine-generated representation of a MATLAB object
% and its children.  Note that handle values may change when these
% objects are re-created. This may cause problems with some callbacks.
% The command syntax may be supported in the future, but is currently
% incomplete and subject to change.
%
% To re-open this system, just type the name of the m-file at the MATLAB
% prompt. The M-file and its associtated MAT-file must be on your path.

% GUI screen to see while plots are being generated

% Written for JANRAD version 6.0 by LT David A. Heathorn

load stby_scrn

global H_STBY_SCRN

H_STBY_SCRN = figure('Color',[0.8 0.8 0.8], ...
   'Colormap',mat0, ...
   'Name','Standby Screen', ...
      'NumberTitle','off', ...
      'PointerShapeCData',mat1, ...
      'Position',[236 243 394 201], ...
   'Tag','Fig2');
b = uicontrol('Parent' ,H_STBY_SCRN, ...
      'Units','points', ...
      'BackgroundColor',[0.752941 0.752941 0.752941], ...
      'Position',[7.5 4.5 278.25 135.75], ...
      'Style','frame', ...
      'Tag','Frame1');


b = uicontrol('Parent' ,H_STBY_SCRN, ...
      'Units','points', ...
      'BackgroundColor',[0.752941 0.752941 0.752941], ...
      'FontSize',25, ...
      'Position',[32.25 95.25 231 37.5], ...
      'String','Please stand-by', ...
      'Style','text', ...
      'Tag','StaticText1');
b = uicontrol('Parent' ,H_STBY_SCRN, ...
      'Units','points', ...
      'BackgroundColor',[0.752941 0.752941 0.752941], ...
      'FontSize',15, ...
      'Position',[33 18.75 231 37.5], ...
      'String','This could take a minute or two', ...
      'Style','text', ...
      'Tag','StaticText1');
```

30. Step_plotter.m

```
% Step_plotter

% Written for JANRAD version 6.0 by LT David A. Heathorn

global Amat Bmat u C  T_STOP T_INC

D=0;
t_start=0;
t_inc=T_INC;
t_stop=T_STOP;

t=t_start:t_inc:t_stop;

if u(1)==1

   for j=1:length(C)
      if C(j,j)==1
         figure
         [num,den]=ss2tf(Amat,Bmat(:,1),C(j,:),D);
         t
         [y]=step(num,den,t);
         plot(t,y)
         xlabel('time (sec)')
         grid
        if j==1
            title('Response of x-velocity (u) to Longitudinal Cyclic
Step Input')
            ylabel('Velocity (ft/sec)')
         elseif j==2
            title('Response of z-velocity (w) to Longitudinal Cyclic
Step Input')
            ylabel('Velocity (ft/sec)')
         elseif j==3
            title('Response of Pitch Rate (q) to Longitudinal Cyclic
Step Input')
            ylabel('Pitch Rate (rad/sec)')
         elseif j==4
            title('Response of Pitch Angle (theta) to Longitudinal
Cyclic Step Input')
            ylabel('Pitch Angle (rad.)')
         elseif j==5
            title('Response of y-veloctiy (v) to Longitudinal Cyclic
Step Input')
            ylabel('Velocity (ft/sec)')
         elseif j==6
            title('Response of Roll Rate (p) to Longitudinal Cyclic Step
Input')
            ylabel('Roll Rate (rad/sec)')
         elseif j==7
            title('Response of Roll Angle (phi) to Longitudinal Cyclic
Step Input')
            ylabel('Roll Angle (rad)')
         elseif j==8
            title('Response of Yaw Rate (r) to Longitudinal Cyclic Step
Input')
```

```
            ylabel('Yaw Rate (rad/sec)')
         elseif j==9
            title('Response of Yaw Angle (psi) to Longitudinal Cyclic
Step Input')
            ylabel('Yaw Angle(rad)')
         end
      end
   end

elseif u(2)==1

   for j=1:length(C)
      if C(j,j)==1
         figure
         [num,den]=ss2tf(Amat,Bmat(:,1),C(j,:),D);
         [y]=step(num,den,t);
         plot(t,y)
         xlabel('time (sec)')
         grid
         xlabel('time (sec)')
         if j==1
            title('Response of x-velocity (u) to Collective Step Input')
            ylabel('Velocity (ft/sec)')
         elseif j==2
            title('Response of z-velocity (w) to Collective Step Input')
            ylabel('Velocity (ft/sec)')
         elseif j==3
            title('Response of Pitch Rate (q) to Collective Step Input')
            ylabel('Pitch Rate (rad/sec)')
         elseif j==4
            title('Response of Pitch Angle (theta) to Collective Step
Input')
            ylabel('Pitch Angle (rad.)')
         elseif j==5
            title('Response of y-veloctiy (v) to Collective Step Input')
            ylabel('Velocity (ft/sec)')
         elseif j==6
            title('Response of Roll Rate (p) to Collective Step Input')
            ylabel('Roll Rate (rad/sec)')
         elseif j==7
            title('Response of Roll Angle (phi) to Collective Step
Input')
            ylabel('Roll Angle (rad)')
         elseif j==8
            title('Response of Yaw Rate (r) to Collective Step Input')
            ylabel('Yaw Rate (rad/sec)')
         elseif j==9
            title('Response of Yaw Angle (psi) to Collective Step
Input')
            ylabel('Yaw Angle(rad)')
         end
      end
   end


elseif u(3)==1
   for j=1:length(C)
```

```
      if C(j,j)==1
         figure
         [num,den]=ss2tf(Amat,Bmat(:,1),C(j,:),D);
         [y]=step(num,den,t);
         plot(t,y)
         xlabel('time (sec)')
         grid
         if j==1
            title('Response of x-velocity (u) to Lateral Cyclic Step
Input')
            ylabel('Velocity (ft/sec)')
         elseif j==2
            title('Response of z-velocity (w) to Lateral Cyclic Step
Input')
            ylabel('Velocity (ft/sec)')
         elseif j==3
            title('Response of Pitch Rate (q) to Lateral Cyclic Step
Input')
            ylabel('Pitch Rate (rad/sec)')
         elseif j==4
            title('Response of Pitch Angle (theta) to Lateral Cyclic
Step Input')
            ylabel('Pitch Angle (rad.)')
         elseif j==5
            title('Response of y-veloctiy (v) to Lateral Cyclic Step
Input')
            ylabel('Velocity (ft/sec)')
         elseif j==6
            title('Response of Roll Rate (p) to Lateral Cyclic Step
Input')
            ylabel('Roll Rate (rad/sec)')
         elseif j==7
            title('Response of Roll Angle (phi) to Lateral Cyclic Step
Input')
            ylabel('Roll Angle (rad)')
         elseif j==8
            title('Response of Yaw Rate (r) to Lateral Cyclic Step
Input')
            ylabel('Yaw Rate (rad/sec)')
         elseif j==9
            title('Response of Yaw Angle (psi) to Lateral Cyclic Step
Input')
            ylabel('Yaw Angle(rad)')
         end
      end
   end


elseif u(4)==1

   for j=1:length(C)
      if C(j,j)==1
         figure
         [num,den]=ss2tf(Amat,Bmat(:,1),C(j,:),D);
         [y]=step(num,den,t);
         plot(t,y)
```

```
        xlabel('time (sec)')
        grid
        if j==1
           title('Response of x-velocity (u) to Pedal Step Input')
           ylabel('Velocity (ft/sec)')
        elseif j==2
           title('Response of z-velocity (w) to Pedal Step Input')
           ylabel('Velocity (ft/sec)')
        elseif j==3
           title('Response of Pitch Rate (q) to Pedal Step Input')
           ylabel('Pitch Rate (rad/sec)')
        elseif j==4
           title('Response of Pitch Angle (theta) to Pedal Step Input')
           ylabel('Pitch Angle (rad.)')
        elseif j==5
           title('Response of y-veloctiy (v) to Pedal Step Input')
           ylabel('Velocity (ft/sec)')
        elseif j==6
           title('Response of Roll Rate (p) to Pedal Step Input')
           ylabel('Roll Rate (rad/sec)')
        elseif j==7
           title('Response of Roll Angle (phi) to Pedal Step Input')
           ylabel('Roll Angle (rad)')
        elseif j==8
           title('Response of Yaw Rate (r) to Pedal Step Input')
           ylabel('Yaw Rate (rad/sec)')
        elseif j==9
           title('Response of Yaw Angle (psi) to Pedal Step Input')
           ylabel('Yaw Angle(rad)')
        end
     end
  end


end
```

31. structure_stab_input.m

```
% Structure Consruction for JANRAD98 stability_control_input.m

% Created for JANRAD version 6.0 by LT David A. Heathorn

S_STAB_INPUT=struct(...
   'Ib',Ib,...
   'hmd',hmd,...
   'lmd',lmd,...
   'ymd',ymd,...
   'im',im,...
   'htd',htd,...
   'ltd',ltd,...
   'ytd',ytd,...
   'bt',bt,...
   'ct',ct,...
   'Rt',Rt,...
   'at',at,...
   'ohmt',ohmt,...
   'Ibt',Ibt,...
   'delta3',delta3,...
   'theta1t',theta1t,...
   'hvd',hvd,...
   'lvd',lvd,...
   'yvd',yvd,...
   'alplov',alplov,...
   'clvertmax',clvertmax,...
   'qvq',qvq,...
   'av',av,...
   'hhd',hhd,...
   'lhd',lhd,...
   'yhd',yhd,...
   'alploh',alploh,...
   'ih',ih,...
   'ah',ah,...
   'qhq',qhq,...
   'vhv1',vhv1,...
   'detafdalpfh',detafdalpfh,...
   'hwd',hwd,...
   'lwd',lwd,...
   'ywd',ywd,...
   'alplow',alplow,...
   'iw',iw,...
   'aw',aw,...
   'ctw',ctw,...
   'crw',crw,...
   'vwv1',vwv1,...
   'detafdalpfw',detafdalpfw,...
   'zcg',zcg,...
   'xcg',xcg,...
   'ycg',ycg,...
   'Ixx',Ixx,...
   'Iyy',Iyy,...
   'Izz',Izz,...
   'Ixz',Ixz,...
   'vfv1',vfv1,...
```

```
'htnd',htnd,...
'ltnd',ltnd,...
'ytnd',ytnd,...
'dian',dian,...
'swirl',swirl,...
'Ytmaxn',Ytmaxn,...
'lttnd',lttnd,...
'db1mddele',db1mddele,...
'da1mddela',da1mddela,...
'dthetomddelc',dthetomddelc,...
'dthetotddelp',dthetotddelp,...
'sidearm',sidearm,...
'maxr',maxr);
```

32. structure_stab_input_1.m

```
% Structure Consruction for JANRAD98 stability_control_input_1.m

% Written for JANRAD version 6.0 by LT David A. Heathorn

S_STAB_INPUT_1=struct(...
    'Ib',Ib,...
    'hmd',hmd,...
    'lmd',lmd,...
    'ymd',ymd,...
    'im',im,...
    'hvd',hvd,...
    'lvd',lvd,...
    'yvd',yvd,...
    'alplov',alplov,...
    'clvertmax',clvertmax,...
    'qvq',qvq,...
    'av',av,...
    'hhd',hhd,...
    'lhd',lhd,...
    'yhd',yhd,...
    'alploh',alploh,...
    'ih',ih,...
    'ah',ah,...
    'qhq',qhq,...
    'vhv1',vhv1,...
    'detafdalpfh',detafdalpfh,...
    'htd',htd,...
    'ltd',ltd,...
    'ytd',ytd,...
    'bt',bt,...
    'ct',ct,...
    'Rt',Rt,...
    'at',at,...
    'ohmt',ohmt,...
    'Ibt',Ibt,...
    'delta3',delta3,...
    'theta1t',theta1t);
```

33. structure_stab_input_2.m

```
% Structure Consruction for JANRAD98 stability_control_input_2.m

% Written for JANRAD version 6.0 by LT David A. Heathorn

S_STAB_INPUT_2=struct(...
    'hwd',hwd,...
    'lwd',lwd,...
    'ywd',ywd,...
    'alplow',alplow,...
    'iw',iw,...
    'aw',aw,...
    'ctw',ctw,...
    'crw',crw,...
    'vwv1',vwv1,...
    'detafdalpfw',detafdalpfw,...
    'zcg',zcg,...
    'xcg',xcg,...
    'ycg',ycg,...
    'Ixx',Ixx,...
    'Iyy',Iyy,...
    'Izz',Izz,...
    'Ixz',Ixz,...
    'vfv1',vfv1,...
    'htnd',htnd,...
    'ltnd',ltnd,...
    'ytnd',ytnd,...
    'dian',dian,...
    'swirl',swirl,...
    'Ytmaxn',Ytmaxn,...
    'lttnd',lttnd,...
    'db1mddele',db1mddele,...
    'da1mddela',da1mddela,...
    'dthetomddelc',dthetomddelc,...
    'dthetotddelp',dthetotddelp,...
    'sidearm',sidearm,...
    'maxr',maxr);
```

34. tail_warn.m

```
function tail_warn()
% This is the machine-generated representation of a MATLAB object
% and its children.  Note that handle values may change when these
% objects are re-created. This may cause problems with some callbacks.
% The command syntax may be supported in the future, but is currently
% incomplete and subject to change.
%
% To re-open this system, just type the name of the m-file at the MATLAB
% prompt. The M-file and its associated MAT-file must be on your path.

% GUI screen to indicated version is not configured for fan-in-tail or
NOTAR

% Created for JANRAD version 6.0 by LT David A. Heathorn

load tail_warn

a = figure('Color',[0.8 0.8 0.8], ...
      'Colormap',mat0, ...
      'Name','Tail Warning', ...
      'NumberTitle','off', ...
      'PointerShapeCData',mat1, ...
      'Position',[252 240 402 201], ...
      'Tag','Fig2');
b = uicontrol('Parent',a, ...
      'Units','points', ...
      'BackgroundColor',[0.752941 0.752941 0.752941], ...
      'FontSize',18, ...
      'Position',[42.75 39.75 212.25 69.75], ...
      'String','JANRAD is not currently configured for fan-in-tail or
NOTAR systems.', ...
      'Style','text', ...
      'Tag','StaticText1');
```

35. temp_stab.m

```
% Structure Consruction for JANRAD98 stability_control_input.m

% JANRAD 98 VERSION 5.0

S_STAB_INPUT=struct(...
    'Ib',Ib,...
    'hmd',hmd,...
    'lmd',lmd,...
    'ymd',ymd,...
    'im',im,...
    'htd',htd,...
    'ltd',ltd,...
    'ytd',ytd,...
    'bt',bt,...
    'ct',ct,...
    'Rt',Rt,...
    'at',at,...
    'ohmt',ohmt,...
    'Ibt',Ibt,...
    'delta3',delta3,...
    'theta1t',theta1t,...
    'hvd',hvd,...
    'lvd',lvd,...
    'yvd',yvd,...
    'alplov',alplov,...
    'clvertmax',clvertmax,...
    'qvq',qvq,...
    'av',av,...
    'hhd',hhd,...
    'lhd',lhd,...
    'yhd',yhd,...
    'alploh',alploh,...
    'ih',ih,...
    'ah',ah,...
    'qhq',qhq,...
    'vhv1',vhv1,...
    'detafdalpfh',detafdalpfh,...
    'hwd',hwd,...
    'lwd',lwd,...
    'ywd',ywd,...
    'alplow',alplow,...
    'iw',iw,...
    'aw',aw,...
    'ctw',ctw,...
    'crw',crw,...
    'vwv1',vwv1,...
    'detafdalpfw',detafdalpfw,...
    'zcg',zcg,...
    'xcg',xcg,...
    'ycg',ycg,...
    'Ixx',Ixx,...
    'Iyy',Iyy,...
    'Izz',Izz,...
    'Ixz',Ixz,...
    'vfv1',vfv1,...
```

```
'htnd',htnd,...
'ltnd',ltnd,...
'ytnd',ytnd,...
'dian',dian,...
'swirl',swirl,...
'Ytmaxn',Ytmaxn,...
'lttnd',lttnd,...
'db1mddele',db1mddele,...
'da1mddela',da1mddela,...
'dthetomddelc',dthetomddelc,...
'dthetotddelp',dthetotddelp,...
'sidearm',sidearm,...
'maxr',maxr);
```

36. time_freq_resp.m

```
function time_freq_resp()
% This is the machine-generated representation of a MATLAB object
% and its children.  Note that handle values may change when these
% objects are re-created. This may cause problems with some callbacks.
% The command syntax may be supported in the future, but is currently
% incomplete and subject to change.
%
% To re-open this system, just type the name of the m-file at the MATLAB
% prompt. The M-file and its associated MAT-file must be on your path.\

% Created for JANRAD version 6.0 by LT David A. Heathorn

load time_freq_resp

global C u type_anal  Amat Bmat H_TF_RESP H_STAB_OUT T_STOP T_INC

type_anal=zeros(4,1);
C=zeros(9,9);      .
u=zeros(4,1);


H_TF_RESP = figure('Color',[0.8 0.8 0.8], ...
       'Colormap',mat0, ...
       'Name','Time and Frequency Response', ...
       'NumberTitle','off', ...
       'PointerShapeCData',mat1, ...
       'Position',[2 28 798 534], ...
    'Tag','Fig1');

b = uicontrol('Parent',H_TF_RESP, ...
       'Units','points', ...
       'BackgroundColor',[0.752941 0.752941 0.752941], ...
       'Position',[63.75 203.25 222 161.25], ...
       'Style','frame', ...
       'Tag','Frame1');
b = uicontrol('Parent',H_TF_RESP, ...
       'Units','points', ...
       'BackgroundColor',[0.752941 0.752941 0.752941], ...
       'Position',[324 40.5 223.5 324], ...
       'Style','frame', ...
       'Tag','Frame2');
b = uicontrol('Parent',H_TF_RESP, ...
       'Units','points', ...
       'BackgroundColor',[0.752941 0.752941 0.752941], ...
       'Position',[64.5 42 222.75 154.5], ...
       'Style','frame', ...
    'Tag','Frame3');

b = uicontrol('Parent',H_TF_RESP, ...
       'Units','points', ...
       'BackgroundColor',[0.752941 0.752941 0.752941], ...
       'FontSize',18, ...
       'Position',[13.5 375.75 570.75 20.25], ...
       'String','TIME and/or FREQUENCY RESPONSE OF COUPLED SYSTEM', ...
       'Style','text', ...
```

```
      'Tag','StaticText1');
H_LAT = uicontrol('Parent',H_TF_RESP, ...
      'Units','points', ...
      'BackgroundColor',[0.752941 0.752941 0.752941], ...
   'Callback',['global u;',...
       'if get(gcbo, ''Value'')==1,'...
          'u=[0; 0; 1; 0];'...
         'set(H_LONG,''Enable'',''off''),',...
          'set(H_COL,''Enable'',''off''),',...
          'set(H_PED,''Enable'',''off''),',...
       'else,',...
          'set(H_LONG,''Enable'',''on''),',...
          'set(H_COL,''Enable'',''on''),',...
          'set(H_PED,''Enable'',''on''),',...
       'end,'],...
      'Position',[94.5 309 87 21.75], ...
      'String','Lateral Cyclic', ...
      'Style','checkbox', ...
      'Tag','Checkbox1');
H_LONG = uicontrol('Parent',H_TF_RESP, ...
      'Units','points', ...
      'BackgroundColor',[0.752941 0.752941 0.752941], ...
   'Callback',['global u;',...
       'if get(gcbo, ''Value'')==1,'...
          'u=[1; 0; 0; 0];'...
          'set(H_LAT,''Enable'',''off''),',...
          'set(H_COL,''Enable'',''off''),',...
          'set(H_PED,''Enable'',''off''),',...
       'else,',...
          'set(H_LAT,''Enable'',''on''),',...
          'set(H_COL,''Enable'',''on''),',...
          'set(H_PED,''Enable'',''on''),',...
       'end,'],...
      'Position',[94.5 278.25 87 21.75], ...
      'String','Longitudinal Cyclic', ...
      'Style','checkbox', ...
      'Tag','Checkbox1');
H_COL = uicontrol('Parent',H_TF_RESP, ...
      'Units','points', ...
      'BackgroundColor',[0.752941 0.752941 0.752941], ...
   'Callback',['global u;',...
       'if get(gcbo, ''Value'')==1,'...
          'u=[0; 1; 0; 0];'...
          'set(H_LAT,''Enable'',''off''),',...
          'set(H_LONG,''Enable'',''off''),',...
          'set(H_PED,''Enable'',''off''),',...
       'else,',...
          'set(H_LAT,''Enable'',''on''),',...
          'set(H_LONG,''Enable'',''on''),',...
          'set(H_PED,''Enable'',''on''),',...
       'end,'],...
      'Position',[93.75 247.5 87 21.75], ...
      'String','Collective', ...
      'Style','checkbox', ...
      'Tag','Checkbox1');
H_PED = uicontrol('Parent',H_TF_RESP, ...
      'Units','points', ...
```

```
      'BackgroundColor',[0.752941 0.752941 0.752941], ...
   'Callback',['global u;',...
       'if get(gcbo, ''Value'')==1,'...
          'u=[0; 0; 0; 1];'...
          'set(H_LAT,''Enable'',''off''),',...
          'set(H_COL,''Enable'',''off''),',...
          'set(H_LONG,''Enable'',''off''),',...
       'else,',...
          'set(H_LAT,''Enable'',''on''),',...
          'set(H_COL,''Enable'',''on''),',...
          'set(H_LONG,''Enable'',''on''),',...
       'end,'],...
      'Position',[94.5 214.5 87 21.75], ...
      'String','Pedal', ...
      'Style','checkbox', ...
   'Tag','Checkbox1');

b = uicontrol('Parent',H_TF_RESP, ...
      'Units','points', ...
      'BackgroundColor',[0.752941 0.752941 0.752941], ...
      'FontSize',14, ...
      'Position',[94.5 174 180 18.75], ...
      'String','Type of Analysis', ...
      'Style','text', ...
   'Tag','StaticText2');
b = uicontrol('Parent',H_TF_RESP, ...
      'Units','points', ...
      'BackgroundColor',[0.752941 0.752941 0.752941], ...
      'Callback',['global type_anal ;'...
      'if get(gcbo, ''Value'')==1,'...
          'type_anal(1)=1;'...
      'elseif get (gcbo, ''Value'')==0,',...
          'type_anal(1)=0;',...
      'end,'],...
      'Position',[94.5 141.75 180 21.75], ...
      'String','Time response to unit step', ...
   'Style','checkbox', ...
   'Tag','Checkbox1');
b = uicontrol('Parent',H_TF_RESP, ...
      'Units','points', ...
      'BackgroundColor',[0.752941 0.752941 0.752941], ...
      'Callback',['global type_anal ;'...
      'if get(gcbo, ''Value'')==1,'...
          'type_anal(2)=1;'...
      'elseif get (gcbo, ''Value'')==0,',...
          'type_anal(2)=0;',...
      'end,'],...
      'Position',[94.5 112.5 179.25 21.75], ...
      'String','Time response to unit impulse', ...
      'Style','checkbox', ...
      'Tag','Checkbox1');
b = uicontrol('Parent',H_TF_RESP, ...
      'Units','points', ...
      'BackgroundColor',[0.752941 0.752941 0.752941], ...
   'Callback',['global type_anal ;'...
      'if get(gcbo, ''Value'')==1,'...
          'type_anal(3)=1;'...
```

```
      'elseif get (gcbo, ''Value'')==0,',...
          'type_anal(3)=0;',...
      'end,'],...
      'Position',[94.5 81.75 181.5 21.75], ...
      'String','Bode plots', ...
      'Style','checkbox', ...
   'Tag','Checkbox1');
b = uicontrol('Parent',H_TF_RESP, ...
      'Units','points', ...
      'BackgroundColor',[0.752941 0.752941 0.752941], ...
      'Callback',['global type_anal ;'...
      'if get(gcbo, ''Value'')==1,'...
          'type_anal(4)=1;'...
      'elseif get (gcbo, ''Value'')==0,',...
          'type_anal(4)=0;',...
      'end,'],...
      'Position',[94.5 51.75 181.5 21.75], ...
      'String','Eigen values', ...
      'Style','checkbox', ...
      'Tag','Checkbox1');

b = uicontrol('Parent',H_TF_RESP, ...
      'Units','points', ...
      'BackgroundColor',[0.752941 0.752941 0.752941], ...
      'FontSize',14, ...
      'Position',[94.5 340.5 180 18.75], ...
      'String','Input Channel', ...
      'Style','text', ...
      'Tag','StaticText2');
b = uicontrol('Parent',H_TF_RESP, ...
      'Units','points', ...
      'BackgroundColor',[0.752941 0.752941 0.752941], ...
      'FontSize',14, ...
      'Position',[339.75 340.5 180 18.75], ...
      'String','Response', ...
      'Style','text', ...
   'Tag','StaticText2');

b = uicontrol('Parent',H_TF_RESP, ...
      'Units','points', ...
      'BackgroundColor',[0.752941 0.752941 0.752941], ...
      'Callback',['global C ;'...
      'if get(gcbo, ''Value'')==1,'...
          'C(1,1)=1;'...
      'elseif get (gcbo, ''Value'')==0,',...
          'C(1,1)=0;',...
      'end,'],...
      'Position',[338.25 309.75 87 21.75], ...
      'String','u', ...
      'Style','checkbox', ...
   'Tag','Checkbox1');
b = uicontrol('Parent',H_TF_RESP, ...
      'Units','points', ...
      'BackgroundColor',[0.752941 0.752941 0.752941], ...
      'Callback',['global C ;'...
      'if get(gcbo, ''Value'')==1,'...
          'C(5,5)=1;'...
```

```
        'elseif get (gcbo, ''Value'')==0,',...
            'C(5,5)=0;',...
        'end,'],...
        'Position',[339 280.5 87 21.75], ...
        'String','v', ...
        'Style','checkbox', ...
        'Tag','Checkbox1');
b = uicontrol('Parent',H_TF_RESP, ...
        'Units','points', ...
        'BackgroundColor',[0.752941 0.752941 0.752941], ...
    'Callback',['global C ;'...
        'if get(gcbo, ''Value'')==1,'...
            'C(2,2)=1;'...
        'elseif get (gcbo, ''Value'')==0,',...
            'C(2,2)=0;',...
        'end,'],...
        'Position',[339 249 87 21.75], ...
        'String','w', ...
        'Style','checkbox', ...
    'Tag','Checkbox1');
b = uicontrol('Parent',H_TF_RESP, ...
        'Units','points', ...
        'BackgroundColor',[0.752941 0.752941 0.752941], ...
    'Callback',['global C ;'...
        'if get(gcbo, ''Value'')==1,'...
            'C(4,4)=1;'...
        'elseif get (gcbo, ''Value'')==0,',...
            'C(4,4)=0;',...
        'end,'],...
        'Position',[339 216.75 87 21.75], ...
        'String','Pitch', ...
        'Style','checkbox', ...
        'Tag','Checkbox1');
b = uicontrol('Parent',H_TF_RESP, ...
        'Units','points', ...
        'BackgroundColor',[0.752941 0.752941 0.752941], ...
        'Callback',['global C ;'...
        'if get(gcbo, ''Value'')==1,'...
            'C(7,7)=1;'...
        'elseif get (gcbo, ''Value'')==0,',...
            'C(7,7)=0;',...
        'end,'],...
        'Position',[339 186.75 87 21.75], ...
        'String','Roll', ...
        'Style','checkbox', ...
        'Tag','Checkbox1');
b = uicontrol('Parent',H_TF_RESP, ...
        'Units','points', ...
        'BackgroundColor',[0.752941 0.752941 0.752941], ...
        'Callback',['global C ;'...
        'if get(gcbo, ''Value'')==1,'...
            'C(9,9)=1;'...
        'elseif get (gcbo, ''Value'')==0,',...
            'C(9,9)=0;',...
        'end,'],...
        'Position',[339 156.75 87 21.75], ...
        'String','Yaw', ...
```

```
        'Style','checkbox', ...
        'Tag','Checkbox1');
b = uicontrol('Parent',H_TF_RESP, ...
        'Units','points', ...
        'BackgroundColor',[0.752941 0.752941 0.752941], ...
        'Callback',['global C ;'...
        'if get(gcbo, ''Value'')==1,'...
            'C(3,3)=1;'...
        'elseif get (gcbo, ''Value'')==0,',...
            'C(3,3)=0;',...
        'end,'],...
        'Position',[339 124.5 87 21.75], ...
        'String','Pitch Rate', ...
        'Style','checkbox', ...
        'Tag','Checkbox1');
b = uicontrol('Parent',H_TF_RESP, ...
        'Units','points', ...
        'BackgroundColor',[0.752941 0.752941 0.752941], ...
        'Callback',['global C ;'...
        'if get(gcbo, ''Value'')==1,'...
            'C(6,6)=1;'...
        'elseif get (gcbo, ''Value'')==0,',...
            'C(6,6)=0;',...
        'end,'],...
        'Position',[339 93.75 87 21.75], ...
        'String','Roll Rate', ...
        'Style','checkbox', ...
        'Tag','Checkbox1');
b = uicontrol('Parent',H_TF_RESP, ...
        'Units','points', ...
        'BackgroundColor',[0.752941 0.752941 0.752941], ...
        'Callback',['global C ;'...
        'if get(gcbo, ''Value'')==1,'...
            'C(8,8)=1;'...
        'elseif get (gcbo, ''Value'')==0,',...
            'C(8,8)=0;',...
        'end,'],...
        'Position',[339 62.25 87 21.75], ...
        'String','Yaw Rate', ...
        'Style','checkbox', ...
    'Tag','Checkbox1');

b = uicontrol('Parent',H_TF_RESP, ...
        'Units','points', ...
    'BackgroundColor',[1 1 1], ...
    'Callback',['global T_INC ;'...
        'T_INC=get(gcbo,''String'');T_INC=str2num(T_INC);'], ...
    'Position',[443.25 200.25 63 15], ...
    'String',T_INC, ...
        'Style','edit', ...
        'Tag','EditText1');
b = uicontrol('Parent',H_TF_RESP, ...
        'Units','points', ...
    'BackgroundColor',[1 1 1], ...
    'Callback',['global T_STOP ;'...
        'T_STOP=get(gcbo,''String'');T_STOP=str2num(T_STOP);'], ...
    'Position',[443.25 252 63 15], ...
```

```
    'String',T_STOP, ...
    'Style','edit', ...
        'Tag','EditText1');
b = uicontrol('Parent',H_TF_RESP, ...
        'Units','points', ...
        'BackgroundColor',[0.752941 0.752941 0.752941], ...
        'Position',[440.25 267.75 69 17.25], ...
        'String','Stop Time (sec)', ...
        'Style','text', ...
        'Tag','StaticText3');
b = uicontrol('Parent',H_TF_RESP, ...
        'Units','points', ...
        'BackgroundColor',[0.752941 0.752941 0.752941], ...
        'Position',[438.75 215.25 69 17.25], ...
        'String','Increment (sec)', ...
        'Style','text', ...
        'Tag','StaticText3');

b = uicontrol('Parent',H_TF_RESP, ...
        'Units','points', ...
    'Callback','global Amat Bmat C u T_STOP T_INC
type_anal;time_freq_resp_fcn cont', ...
        'FontSize',14, ...
        'Position',[311.25 5.25 105.75 27], ...
        'String','CONTINUE >>', ...
        'Tag','Pushbutton1');
b = uicontrol('Parent',H_TF_RESP, ...
    'Units','points', ...
    'Callback','global Amat Bmat C u T_STOP T_INC
type_anal;time_freq_resp_fcn back', ...
        'FontSize',14, ...
        'Position',[177 5.25 105.75 27], ...
        'String','<< BACK', ...
    'Tag','Pushbutton1');


assignin('base','H_LONG',H_LONG);
assignin('base','H_LAT',H_LAT);
assignin('base','H_COL',H_COL);
assignin('base','H_PED',H_PED);
```

37. time_freq_resp_fcn.m

```
function time_freq_resp_fcn(Action)

% Written for JANRAD version 6.0 by LT David A. Heathorn

global Amat Bmat u C type_anal time_vec H_STBY_SCRN H_TF_RESP H_STAB_OUT
T_INC T_STOP


if isempty(nargin)
   return
end

   switch Action
   case 'cont'

      stby_scrn
      pause(1)

      if type_anal(1)==1
         step_plotter
      end

      if type_anal(2)==1
         imp_plotter
      end

      if type_anal(3)==1
         bode_plotter
      end

      if type_anal(4)==1
         eigen_plotter
      end

      close(H_STBY_SCRN)

   case 'back'
      stab_out_1
      close(H_TF_RESP)


         end
```

38. Trim.m

```
% Trim.m

%     Trim routine for collective/cyclic.
%     JANRAD 98 VERSION 5.0
% Modified for JAJRAD version 6.0 by LT David A. Heathorn

global RADSPC_VAL NL_TWIST_VAL NEW_AUX_VAL FIX_TPP_VAL NEW_TPP
lamdaT_trim

if ~exist('STAB_ON')

if get(H_AS,'Value')==1
   IT_PARAM='AIRSPEED';
   IT_UNIT='KTS';
elseif get(H_AL,'Value')==1
   IT_PARAM='ALTITUDE';
   IT_UNIT='FT';
elseif get(H_GW,'Value')==1
   IT_PARAM='GROSS WEIGHT';
   IT_UNIT='LBS';
elseif get(H_BT,'Value')==1
   IT_PARAM='BLADE TWIST';
   IT_UNIT='DEG';
elseif get(H_BTR,'Value')==1
   IT_PARAM='BLADE TAPER RATIO';
   IT_UNIT='';
elseif get(H_SOT,'Value')==1
   IT_PARAM='START OF TAPER';
   IT_UNIT='FT';
elseif get(H_WSA,'Value')==1
   IT_PARAM='WING SPAN AREA';
   IT_UNIT='FT^2';
elseif get(H_RBR,'Value')==1
   IT_PARAM='ROTOR BLADE RADIUS';
   IT_UNIT='FT';
elseif get(H_RBS,'Value')==1
   IT_PARAM='ROTOR BLADE SPEED';
   IT_UNIT='RAD/SEC';
end
end
% Following was added to adjust flight conditions for UH-60A runs
%if Vinf/1.68781 < 50
%   Afh=35;
%   PA=2500;
%   temp=68;
%   GW=17377;
%   CLhoriz=0;
%elseif Vinf/1.68781 <= 90
%   Afh=32;
%   PA=2500;
%   temp=68;
%   GW=17377;
%   CLhoriz=-.1;
%elseif Vinf/1.68781 <= 140
%   Afh=30;
```

```
%   PA=2500;
%   temp=68;
%   GW=17300;
%   CLhoriz=-0.3;
%else
%   Afh=28;
%   PA=2500;
%   temp=68;
%   GW=17000;
%   CLhoriz=-0.5;
%end

set(H_STATUS,'String','EXECUTING ROTOR TRIM ROUTINE')
set(H_STATUS1,'String',['RUN ELAPSED TIME IS ' num2str(fix(toc)) '
SECONDS'])
if ~exist('STAB_ON')
   if get(H_NI,'Value')==1
   set(H_STATUS3,'String','')
   else
   set(H_STATUS3,'String',['ITERATION PARAMETER: ' IT_PARAM ' = '
num2str(itervar) num2str(IT_UNIT)])
   end
end

pause(3)
%   *** calculation of required parameters ***



rho=.002377*(-.000031*PA+(-.002*temp+1.118));



%   *** first guess at rotor profile drag ( H force) ***

if Vinf < 16.9,

  Drotor=0;

else

  Drotor=Vinf*(rho/.002377);

end

q=0.5*rho*Vinf^2;

Adisk=pi*R^2;

Vtip=omega*R;
temp_rank=temp+459.67;
spd_snd=49.1*sqrt(temp_rank);

%  Section added to establish maximum tip speed
%if (Vtip+Vinf)/spd_snd>0.87
%   spd_max=0.87*spd_snd
```

```
%   Vtip=spd_max-Vinf
%   omega=Vtip/R
%end

%  Section added to set wing lift at a certain value and determine
%  the required wing CL       %% This is for compound helos%%
%if Vinf>=160*1.68781
%   perclift= 0.7;
%   Lwing=GW*perclift
%   CLwing= Lwing/(q*Swing)
%if Vinf>=110*1.68781
%   perclift=.255+.00293*(Vinf-(110*1.68781))
%   Lwing=GW*perclift
%   CLwing= Lwing/(q*Swing)
%else
%  Lwing=q*CLwing*Swing;

%end
Lwing=q*CLwing*Swing;

Dfuse=q*Afh;

CDwing=CDowing+(CLwing^2/(ewing*pi*(bwing^2/Swing)));

CDhoriz=CDohoriz+(CLhoriz^2/(.8*pi*(bhoriz^2/Shoriz)));

CDvert=CDovert+(CLvert^2/(.8*pi*(bvert^2/Svert)));

Dwing=q*CDwing*Swing;

Dhoriz=q*CDhoriz*Shoriz;

Dvert=q*CDvert*Svert;
Dftotal=(Dfuse+Dwing+Dhoriz+Dvert);

% This section is for compound helos, adjustment of aux thrust
efficiency with airspeed
if Vinf/1.68781<=70
   AUXEFF=.650;
elseif Vinf/1.68781<=100
   AUXEFF=.65+.0025*((Vinf/1.68781)-70);
elseif Vinf/1.68781<=160
   AUXEFF=.725+.0025*((Vinf/1.68781)-100);
elseif Vinf/1.68781<=210
   AUXEFF=.85+.00007*((Vinf/1.68781)-160);
else
   AUXEFF=.847;
end

% This section provided aux thrust schedule for compound helo
switch NEW_AUX_VAL
case 0
   Dftotal=(Dfuse+Dwing+Dhoriz+Dvert)-Taux;
case 1
   if PA==8000
      Taux=16*Vinf/1.68781;  % linear increase in aux thrust up to 210
knots
```

```
      S_PERF_INPUT.Taux=Taux;
      S_USER_INPUT.Taux=Taux;
   elseif PA==5000
      Taux=17.024*Vinf/1.68781;  % linear increase in aux thrust up to
210 knots
      S_PERF_INPUT.Taux=Taux;
      S_USER_INPUT.Taux=Taux;
   elseif PA==0
      Taux=Dftotal;
      S_PERF_INPUT.Taux=Taux;
      S_USER_INPUT.Taux=Taux;
   end
end
Lhoriz=q*CLhoriz*Shoriz;

Lvert=q*CLvert*Svert;

Lftotal=Lwing+Lhoriz;
if FIX_TPP_VAL==1
   alphaT=NEW_TPP;      %set tip path angle
else
   alphaT=atan2((Dftotal+Drotor),(GW-Lftotal));
end
mu=Vinf*cos(alphaT)/Vtip;



%   *** thrust calculation ***



if Vinf < 16.9,
   T=(1+(0.4*Afv/Adisk))*GW;
else

   T=(GW-Lftotal)/cos(alphaT);
end

CT=T/(Adisk*rho*Vtip^2);

  %Values to check output
%   *** setup blade radius elements, azimuth elements,

%       induced velocity distributions, and determination

%       of coning angle and tip loss parameter ***


B=1-(sqrt(2*CT)/b);

Reff=B*R;

Rbar=Reff-e;


if RADSPC_VAL==1
   NEW_r1=[NEW_r, Reff/R];
```

```
   n=length(NEW_r1);
   dr=diff(NEW_r1)*R;
   r=(NEW_r1(1:n-1)*R)+dr/2;
else
   dr=(Reff-grip)/nbe;
   r=grip:dr:Reff-dr;,r=r+dr/2;
end
if NL_TWIST_VAL==1
   NL_TWIST=NL_TWIST/57.3;
   n=length(NL_TWIST);
   if RADSPC_VAL==1
      y=((Reff/R)-NEW_r(n))*((NL_TWIST(n)-NL_TWIST(n-1))/(NEW_r(n)-
NEW_r(n-1)));
   else
      y=((Reff/R)-r(n))*((NL_TWIST(n)-NL_TWIST(n-1))/(r(n)- r(n-1)));
   end
   NL_TWIST1=[NL_TWIST (NL_TWIST(n)+y)];
   m=length(NL_TWIST1);
   dTW=diff(NL_TWIST1);
   twist=(NL_TWIST1(1:m-1))+dTW/2;
   betat=twist;
else
   betat=twist*(0.7-(r/R));

end
rT1=0.7;,% *** first guess at rT ***

RbarT=rT1*Rbar;

mblade=wblade/32.17;

betao=asin((T/b*RbarT-(.5*(R-e)+e)*wblade)/((.5*(R-
e)+e)^2*omega^2*mblade));

psi=0:360/naz:360-360/naz;,psi=psi'/57.3;



%% set up vector of blade element chords and then varies them as

%%  requested with the blade taper and blade taper start position

%%  rchord=root chord

%%  cblade=vector of blade element chord lengths

%%  tr=taper ratio (tip/root)

%%  trst=taper ratio start position (r/R)


cblade=rchord*ones(size(r));  % gives all elements same chord length
initially
```

```
if tr==0    % prevents division by zero later in code
    tr=1;   %  in case 0 is enter for taper ratio instead
end         %  of 1 for no taper


if trst==0
    slope=(rchord-rchord*tr)/(Reff-grip); % Modifies each element
    cblade=cblade-slope*(r-grip);          %  chord length wrt input
    tchord=cblade(nbe);                    %  taper ratio which has been
    mchord=sum(cblade)/nbe;                %  been converted into a slope
                                     %  top portion takes into
else                                 %  account the possibility that
    slope=(rchord-rchord*tr)/(R*(1-trst));      %  a 0 start position is
really at
    z=fix(nbe*trst);                       %  the start of the aero
portion
      if z<=1           % prevents beginning index fm being zero
        z=1;
      end
    cblade(z:nbe)=cblade(z:nbe)-(r(z:nbe)-r(z))*slope;
    tchord=cblade(nbe);
    mchord=sum(cblade)/nbe;
end



%   *** induced velocity determination ***


  if Vinf < 16.9,
    A=4*pi;
    Bv=(b/2)*omega*a.*cblade;
```

```
    Tv=0;

    delT=T-Tv;

    while abs(delT) > .01*T   % Prouty Eqns for Hover

      thetav=betat+thetao;

      C=(-b/2).*cblade*omega^2.*r*a.*thetav;

      vi=(-Bv+sqrt(Bv.^2-(4*A*C)))/(2*A);

      dTv=(b/2)*rho*((omega*r).^2)*a.*(thetav-
(vi./(omega*r))).*cblade.*dr;
      Tv=sum(dTv);

      delT=T-Tv;

        if delT < 0,

          thetao=thetao-0.5*thetao*abs(delT/T);

        else

          thetao=thetao+0.5*thetao*abs(delT/T);

        end

     end
  else                    % Wheatley Eqn for Fwd flt

     lamdaT=0;

     lamda=1;

     while abs(lamdaT-lamda)>1e-4

              lamda=lamdaT;

              lamdaT=mu*sin(alphaT)+0.5*CT/sqrt(lamdaT^2+mu^2);

     end


    vi=lamdaT*Vtip-Vinf*sin(alphaT);

    vi=vi*ones(size(r));

  end



%   *** first guess at theta ***
```

```
theta1c=0.035*((0.0006e-3*Vinf^2+0.244e-3*Vinf)/0.105);

theta1s=-0.087*((0.0006e-3*Vinf^2+0.244e-3*Vinf)/0.105);

theta=thetao+theta1c.*cos(psi)+theta1s.*sin(psi);



%   *** rotor trimming routine ***


set(H_STATUS,'String','TRIMMING COLLECTIVE')
set(H_STATUS1,'String',['RUN ELAPSED TIME IS ' num2str(fix(toc)) '
SECONDS'])
pause(3)
set(H_STATUS2,'String','')


k=1;

error0=(T*.02)+1;


  while abs(error0) > T*.02
    set(H_STATUS2,'String',['COLLECTIVE TRIM ROUTINE IS ON ITERATION #
',num2str(k)])
    set(H_STATUS1,'String',['RUN ELAPSED TIME IS ' num2str(fix(toc)) '
SECONDS'])
    Tpsi=zeros(size(psi));
    Npsi=zeros(size(psi));
    thrcalc
    if k>1,   % Eccles change: These three lines were added.

     error1;

    end

    error0=T-(mean(Tpsi)*b);

      if error0 < -T*.02,

        thetao=thetao-0.35*thetao*abs(1.5*error0/T)*(1-mu);

      elseif error0 > T*.02,

        thetao=thetao+0.35*thetao*abs(1.5*error0/T)*(1-mu);

      end
```

```
    theta=thetao+theta1c.*cos(psi)+theta1s.*sin(psi);

    if k > 1,

      if abs(error0) > abs(error1),

        clc

        trim_warning
        set(H_GO,'Enable','off');
        set(H_RES,'Enable','off');
        set(H_RUPT,'Enable','off');
        set(H_BK,'Enable','on');
        error('*** Did Not Trim ***')
      end

   end
   error1=error0;
   k=k+1;
 end


set(H_STATUS,'String','TRIMMING CYCLIC')
set(H_STATUS1,'String',['RUN ELAPSED TIME IS ' num2str(fix(toc)) '
SECONDS'])
set(H_STATUS2,'String','')
pause(3)


t0=clock;

k=1;

error0=(((T/b)*rT1*(R-grip))*.04)+1;

  while error0 > ((T/b)*rT1*(R-grip))*.04

    set(H_STATUS2,'String',['CYCLIC TRIM ROUTINE IS ON ITERATION #
',num2str(k)])

    set(H_STATUS1,'String',['RUN ELAPSED TIME IS ' num2str(fix(toc)) '
SECONDS'])
    time=etime(clock,t0);

    if time > 15,

      set(H_STATUS,'String','STILL TRIMMING ...')
      set(H_STATUS1,'String',['RUN ELAPSED TIME IS ' num2str(fix(toc)) '
SECONDS'])
            set(H_STATUS2,'String',['CYCLIC TRIM ROUTINE IS ON ITERATION
# ',num2str(k)])
      pause(2)
      t0=clock;

    end
```

```
    Mpsi(:,k)=zeros(size(psi));
    tmcalc
    theta=[theta theta(:,k)];
    Mpsi=[Mpsi Mpsi(:,k)];

%   *** calculation of initial dthetadM ***

      if k < 2,
        theta(:,k+1)=theta(:,k)+0.25/57.3;
        Mpsi(:,k+1)=zeros(size(psi));
        k=k+1;
        tmcalc
        k=k-1;
        dthetadM=(theta(:,k+1)-theta(:,k))./(Mpsi(:,k+1)-Mpsi(:,k));
      end

%   *** calculation of M first harmonic parameters ***

    M1c=2*sum(Mpsi(:,k).*cos(psi))/naz;
    M1s=2*sum(Mpsi(:,k).*sin(psi))/naz;

%   *** removal of first harmonic terms from Mpsi ***

    Mpsi(:,k+1)=Mpsi(:,k)-M1c.*cos(psi)-M1s.*sin(psi);
    delM=Mpsi(:,k+1)-Mpsi(:,k);
    error0=max(delM)-min(delM);

    if k > 1,
       if error0 > error1,
```

```
      clc
      trim_warning
      set(H_GO,'Enable','off');
      set(H_RES,'Enable','off');
      set(H_RUPT,'Enable','off');
      set(H_BK,'Enable','on');
      error('*** END OF PROGRAM ***')
    end

  end
  error1=error0;
%   *** calculation of new theta ***


  delM=0.5*(1-mu)*delM;

  theta(:,k+1)=theta(:,k)+(dthetadM.*delM);


  if error0 <= ((T/b)*rT1*(R-grip))*.04,

    theta1c=2*sum(theta(:,k).*cos(psi))/naz;

    theta1s=2*sum(theta(:,k).*sin(psi))/naz;

  else

    theta1c=2*sum(theta(:,k+1).*cos(psi))/naz;

    theta1s=2*sum(theta(:,k+1).*sin(psi))/naz;

  end


  theta(:,k+1)=thetao+theta1c.*cos(psi)+theta1s.*sin(psi);

%   *** calculation of new dthetadM ***


  theta=[theta theta(:,k+1)];

  Mpsi=[Mpsi Mpsi(:,k+1)];

  theta(:,k+2)=theta(:,k)+0.25/57.3;

  Mpsi(:,k+2)=zeros(size(Mpsi(:,k+1)));

  k=k+2;

  tmcalc
```

```
    k=k-2;

    dthetadM=(theta(:,k+2)-theta(:,k))./(Mpsi(:,k+2)-Mpsi(:,k));
    k=k+1;

  end

set(H_STATUS,'String','ADJUSTING COLLECTIVE')
set(H_STATUS1,'String',['RUN ELAPSED TIME IS ' num2str(fix(toc)) '
SECONDS'])
set(H_STATUS2,'String','')
pause(3)


theta=theta(:,k);

k=1;

error0=(T*.01)+1;



  while abs(error0) > T*.01

    Tpsi=zeros(size(psi));
    Npsi=zeros(size(psi));
    thrcalc

    error0=T-(mean(Tpsi)*b);

      if error0 < -T*.01,

        thetao=thetao-0.25*thetao*abs(1.25*error0/T)*(1-mu);

      elseif error0 > T*.01,

        thetao=thetao+0.25*thetao*abs(1.25*error0/T)*(1-mu);

      end

    theta=thetao+theta1c.*cos(psi)+theta1s.*sin(psi);

    if k > 1,

      if abs(error0) > abs(error1),

         clc
         trim_warning
         set(H_GO,'Enable','off');
         set(H_RES,'Enable','off');
         set(H_RUPT,'Enable','off');
         set(H_BK,'Enable','on');
         error('*** END OF PROGRAM ***')
      end

    end
```

```
    error1=error0;

    k=k+1;

  end



  % *** calculating drag moments ***
set(H_STATUS2,'String','CALCULATING DRAG MOMENT')
set(H_STATUS1,'String',['RUN ELAPSED TIME IS ' num2str(fix(toc)) '
SECONDS'])
pause(2)
DMpsi=zeros(size(psi));

dmcalc
%   *** calculating rotor H force ***

set(H_STATUS2,'String','CALCULATING ROTOR DRAG')
set(H_STATUS1,'String',['RUN ELAPSED TIME IS ' num2str(fix(toc)) '
SECONDS'])
pause(2)
  if Vinf < 16.9,

    Hrotor=0;

    dT=[dT ddT];
    dN=[dN ddN];

    dD=[dD ddD];

  else

     dT=[dT ddT];
     dN=[dN ddN];
     dD=[dD ddD];

      for i=1:length(r)+1,

        H1c(i)=2*sum(dT(:,i).*cos(psi))/naz;

        H1s(i)=2*sum(dD(:,i).*sin(psi))/naz;

      end

    Hrotor=(((b*cos(alphaT)/2)*(sum(H1s)-
sin(betao)*sum(H1c)))+Drotor)/2;

end




%   *** calculating new rT ***
```

```
rT2=(((mean(Mpsi(:,length(Mpsi(1,:))-1))/mean(Tpsi))/R)+rT1)/2;



%   *** check rotor drag and rT, retrim rotor if required ***


while abs(Drotor-Hrotor) > 0.2*Hrotor | abs(rT1-rT2) > 0.015*rT1


  if abs(Drotor-Hrotor) > 0.2*Hrotor,

     set(H_STATUS,'String','ADJUSTING ROTOR DRAG')
     set(H_STATUS1,'String',['RUN ELAPSED TIME IS ' num2str(fix(toc)) '
SECONDS'])
     set(H_STATUS2,'String',['CURRENT ROTOR DRAG = ' num2str(Drotor) '
LB'])
     pause(3)
  end

  Drotor=Hrotor;

  if abs(rT1-rT2) > 0.015*rT1,

     set(H_STATUS,'String','ADJUSTING MEAN THRUST LOCATION')
     set(H_STATUS1,'String',['RUN ELAPSED TIME IS ' num2str(fix(toc)) '
SECONDS'])
     set(H_STATUS2,'String',['NEW MEAN THRUST LOCATION IS ' num2str(rT2)
' r/R'])
     pause(2)
  end
  set(H_STATUS,'String','RETRIMMING ROTOR')
  set(H_STATUS1,'String',['RUN ELAPSED TIME IS ' num2str(fix(toc)) '
SECONDS'])
  pause(3)
  set(H_STATUS2,'String','')
  dT=dT(:,1:nbe);
  dN=dN(:,1:nbe);

  dD=dD(:,1:nbe);



  %   *** recalculating parameters ***



  if FIX_TPP_VAL==1
     alphaT=NEW_TPP;      %set tip path angle
  else
     alphaT=atan2((Dftotal+Drotor),(GW-Lftotal));
```

```
end

%alphaT80=0;
%if S_USER_INPUT.Vinf<80
%   alphaT=atan2((Dftotal+Drotor),(GW-Lftotal));
%elseif S_USER_INPUT.Vinf==80
%   alphaT=atan2((Dftotal+Drotor),(GW-Lftotal));
%   alphaT80=alphaT;
%elseif S_USER_INPUT.Vinf>=140
%   alphaT=0;
%else
%   alphaT=(1-(((Vinf-135.02479)/100)*.98747))*alphaT80;
%end
mu=Vinf*cos(alphaT)/Vtip;

if Vinf >= 16.9,            % Wheatley Eqn for Fwd flt

   T=(GW-Lftotal)/cos(alphaT);

   CT=T/(Adisk*rho*Vtip^2);

   lamdaT=0;

   lamda=1;

   while abs(lamdaT-lamda)>1e-4

         lamda=lamdaT;

         lamdaT=mu*sin(alphaT)+0.5*CT/sqrt(lamdaT^2+mu^2);
         lamdaT_trim=lamdaT;

   end

   vi=lamdaT*Vtip-Vinf*sin(alphaT);

   vi=vi*ones(size(r));
end



B=1-(sqrt(2*CT)/b);

Reff=B*R;

Rbar=Reff-e;


if RADSPC_VAL==1
   NEW_r1=[NEW_r, Reff/R];
   n=length(NEW_r1);
   dr=diff(NEW_r1)*R;
   r=(NEW_r1(1:n-1)*R)+dr/2;
else
   dr=(Reff-grip)/nbe;
   r=grip:dr:Reff-dr;,r=r+dr/2;
end
```

```
  RbarT=rT2*Rbar;

  betao=asin((T/b*RbarT-(.5*(R-e)+e)*wblade)/((.5*(R-
e)+e)^2*omega^2*mblade));




%   *** trimming collective ***


  t0=clock;

  k=1;

  error0=(T*.02)+1;


  while abs(error0) > T*.02
      set(H_STATUS2,'String',['ROTOR TRIM ROUTINE IS ON ITERATION #
',num2str(k)])

      set(H_STATUS1,'String',['RUN ELAPSED TIME IS ' num2str(fix(toc)) '
SECONDS'])
      Tpsi=zeros(size(psi));
      Npsi=zeros(size(psi));
      thrcalc

      error0=T-(mean(Tpsi)*b);

      if error0 < -T*.02,

        thetao=thetao-0.35*thetao*abs(1.5*error0/T)*(1-mu);

      elseif error0 > T*.02,

        thetao=thetao+0.35*thetao*abs(1.5*error0/T)*(1-mu);

      end

      theta=thetao+theta1c.*cos(psi)+theta1s.*sin(psi);

      if k > 1,

        if abs(error0) > abs(error1),

           clc
           trim_warning
           set(H_GO,'Enable','off');
           set(H_RES,'Enable','off');
           set(H_RUPT,'Enable','off');
           set(H_BK,'Enable','on');
           error('*** END OF PROGRAM ***')
```

```
        end

      end

      error1=error0;
      k=k+1;
   end

%    *** trimming cyclic ***


   k=1;

   error0=(((T/b)*rT2*(R-grip))*.04)+1;


   while error0 > ((T/b)*rT2*(R-grip))*.04

     set(H_STATUS2,'String',['CYCLIC TRIM ROUTINE IS ON ITERATION #
',num2str(k)])
     set(H_STATUS1,'String',['RUN ELAPSED TIME IS ' num2str(fix(toc)) '
SECONDS'])
     time=etime(clock,t0);

     if time > 15,

        set(H_STATUS,'String','STILL TRIMMING ...')
        set(H_STATUS1,'String',['RUN ELAPSED TIME IS ' num2str(fix(toc))
' SECONDS'])
        set(H_STATUS2,'String',['CYCLIC TRIM ROUTINE IS ON ITERATION #
',num2str(k)])

        pause(3)
        t0=clock;

     end


     Mpsi(:,k)=zeros(size(psi));

     tmcalc

     theta=[theta theta(:,k)];

     Mpsi=[Mpsi Mpsi(:,k)];


%    *** calculation of initial dthetadM ***


     if k < 2,
```

```
    theta(:,k+1)=theta(:,k)+0.25/57.3;

    Mpsi(:,k+1)=zeros(size(psi));

    k=k+1;

    tmcalc

    k=k-1;

    dthetadM=(theta(:,k+1)-theta(:,k))./(Mpsi(:,k+1)-Mpsi(:,k));

  end


%   *** calculation of M first harmonic parameters ***


  M1c=2*sum(Mpsi(:,k).*cos(psi))/naz;

  M1s=2*sum(Mpsi(:,k).*sin(psi))/naz;


%   *** removal of first harmonic terms from Mpsi ***


  Mpsi(:,k+1)=Mpsi(:,k)-M1c.*cos(psi)-M1s.*sin(psi);

  delM=Mpsi(:,k+1)-Mpsi(:,k);

  error0=max(delM)-min(delM);

  if k > 1,

    if error0 > error1,

       clc
       trim_warning
       set(H_GO,'Enable','off');
       set(H_RES,'Enable','off');
       set(H_RUPT,'Enable','off');
       set(H_BK,'Enable','on');
       error('*** END OF PROGRAM ***')
    end

  end

  error1=error0;
```

```
%   *** calculation of new theta ***


    delM=0.5*(1-mu)*delM;

    theta(:,k+1)=theta(:,k)+(dthetadM.*delM);

    if error0 <= ((T/b)*rT2*(R-grip))*.04,

      theta1c=2*sum(theta(:,k).*cos(psi))/naz;

      theta1s=2*sum(theta(:,k).*sin(psi))/naz;

    else

      theta1c=2*sum(theta(:,k+1).*cos(psi))/naz;

      theta1s=2*sum(theta(:,k+1).*sin(psi))/naz;

    end

    theta(:,k+1)=thetao+theta1c.*cos(psi)+theta1s.*sin(psi);


%   *** calculation of new dthetadM ***


    theta=[theta theta(:,k+1)];

    Mpsi=[Mpsi Mpsi(:,k+1)];

    theta(:,k+2)=theta(:,k)+0.25/57.3;

    Mpsi(:,k+2)=zeros(size(Mpsi(:,k+1)));

    k=k+2;

    tmcalc

    k=k-2;

    dthetadM=(theta(:,k+2)-theta(:,k))./(Mpsi(:,k+2)-Mpsi(:,k));

    k=k+1;
  end

%   *** retrimming collective ***


  theta=theta(:,k);

  k=1;
```

```
   error0=(T*.01)+1;



   while abs(error0) > T*.01
      set(H_STATUS2,'String',['COLLECTIVE TRIM ROUTINE IS ON ITERATION #
',num2str(k)])
      set(H_STATUS1,'String',['RUN ELAPSED TIME IS ' num2str(fix(toc)) '
SECONDS'])
      Tpsi=zeros(size(psi));
      Npsi=zeros(size(psi));
      thrcalc

      error0=T-(mean(Tpsi)*b);

        if error0 < -T*.01,

          thetao=thetao-0.25*thetao*abs(1.25*error0/T)*(1-mu);

        elseif error0 > T*.01,

          thetao=thetao+0.25*thetao*abs(1.25*error0/T)*(1-mu);

        end

      theta=thetao+theta1c.*cos(psi)+theta1s.*sin(psi);

      if k > 1,

        if abs(error0) > abs(error1),

           clc
           trim_warning
           set(H_GO,'Enable','off');
           set(H_RES,'Enable','off');
           set(H_RUPT,'Enable','off');
           set(H_BK,'Enable','on');
           error('*** END OF PROGRAM ***')
        end

      end

      error1=error0;

      k=k+1;

   end



%   *** recalculating rotor H force ***



   if Vinf < 16.9,
```

```
    Hrotor=0;

    dT=[dT ddT];
    dN=[dN ddN];

    dD=[dD ddD];

  else

    dT=[dT ddT];
    dN=[dN ddN];

    dD=[dD ddD];

      for i=1:length(r)+1,

        H1c(i)=2*sum(dT(:,i).*cos(psi))/naz;

        H1s(i)=2*sum(dD(:,i).*sin(psi))/naz;

      end

    Hrotor=(((b*cos(alphaT)/2)*(sum(H1s)-
sin(betao)*sum(H1c)))+Drotor)/2;

  end




%   *** recalculating rT ***



  rT1=rT2;

  rT2=(((mean(Mpsi(:,length(Mpsi(1,:))-1))/mean(Tpsi))/R)+rT1)/2;

end




% *** recalculating drag moments ***



dT=dT(:,1:nbe);
dN=dN(:,1:nbe);
dD=dD(:,1:nbe);

DMpsi=zeros(size(psi));

dmcalc
```

```
dT=[dT ddT];
dN=[dN ddN];

dD=[dD ddD];

set(H_STATUS,'String','ROTOR TRIMMED')
set(H_STATUS1,'String',['RUN ELAPSED TIME IS ' num2str(fix(toc)) '
SECONDS'])
set(H_STATUS2,'String','IS THIS BETTER THAN JANRAD 3 OR WHAT?')
pause(3)
set(H_STATUS2,'String','')

save perftemp mchord DMpsi  % Eccles addition - perf.m was

                            % not recognizing mchord and DMpsi.
```

39. unstructure_stab_input.m

```
% Unstructure Consruction for JANRAD98 stability_control_input.m

% Written for JANRAD version 6.0 by LT David A. Heathorn

   Ib=S_STAB_INPUT.Ib;
   hmd=S_STAB_INPUT.hmd;
   lmd=S_STAB_INPUT.lmd;
   ymd=S_STAB_INPUT.ymd;
   im=S_STAB_INPUT.im;
   htd=S_STAB_INPUT.htd;
   ltd=S_STAB_INPUT.ltd;
   ytd=S_STAB_INPUT.ytd;
   bt=S_STAB_INPUT.bt;
   ct=S_STAB_INPUT.ct;
   Rt=S_STAB_INPUT.Rt;
   at=S_STAB_INPUT.at;
   ohmt=S_STAB_INPUT.ohmt;
   Ibt=S_STAB_INPUT.Ibt;
   delta3=S_STAB_INPUT.delta3;
   theta1t=S_STAB_INPUT.theta1t;
   hvd=S_STAB_INPUT.hvd;
   lvd=S_STAB_INPUT.lvd;
   yvd=S_STAB_INPUT.yvd;
   alplov=S_STAB_INPUT.alplov;
   clvertmax=S_STAB_INPUT.clvertmax;
   qvq=S_STAB_INPUT.qvq;
   av=S_STAB_INPUT.av;
   hhd=S_STAB_INPUT.hhd;
   lhd=S_STAB_INPUT.lhd;
   yhd=S_STAB_INPUT.yhd;
   alploh=S_STAB_INPUT.alploh;
   ih=S_STAB_INPUT.ih;
   ah=S_STAB_INPUT.ah;
   qhq=S_STAB_INPUT.qhq;
   vhv1=S_STAB_INPUT.vhv1;
   detafdalpfh=S_STAB_INPUT.detafdalpfh;
   hwd=S_STAB_INPUT.hwd;
   lwd=S_STAB_INPUT.lwd;
   ywd=S_STAB_INPUT.ywd;
   alplow=S_STAB_INPUT.alplow;
   iw=S_STAB_INPUT.iw;
   aw=S_STAB_INPUT.aw;
   ctw=S_STAB_INPUT.ctw;
   crw=S_STAB_INPUT.crw;
   vwv1=S_STAB_INPUT.vwv1;
   detafdalpfw=S_STAB_INPUT.detafdalpfw;
   zcg=S_STAB_INPUT.zcg;
   xcg=S_STAB_INPUT.xcg;
   ycg=S_STAB_INPUT.ycg;
   Ixx=S_STAB_INPUT.Ixx;
   Iyy=S_STAB_INPUT.Iyy;
   Izz=S_STAB_INPUT.Izz;
   Ixz=S_STAB_INPUT.Ixz;
   vfv1=S_STAB_INPUT.vfv1;
   htnd=S_STAB_INPUT.htnd;
```

```
ltnd=S_STAB_INPUT.ltnd;
ytnd=S_STAB_INPUT.ytnd;
dian=S_STAB_INPUT.dian;
swirl=S_STAB_INPUT.swirl;
Ytmaxn=S_STAB_INPUT.Ytmaxn;
lttnd=S_STAB_INPUT.lttnd;
db1mddele=S_STAB_INPUT.db1mddele;
da1mddela=S_STAB_INPUT.da1mddela;
dthetomddelc=S_STAB_INPUT.dthetomddelc;
dthetotddelp=S_STAB_INPUT.dthetotddelp;
sidearm=S_STAB_INPUT.sidearm;
maxr=S_STAB_INPUT.maxr;
```

40. unstructure_stab_input_1.m

```
% Unstructure Consruction for JANRAD98 stability_control_input_1.m

% Written for JANRAD version 6.0 by LT David A. Heathorn

   Ib=S_STAB_INPUT_1.Ib;
   hmd=S_STAB_INPUT_1.hmd;
   lmd=S_STAB_INPUT_1.lmd;
   ymd=S_STAB_INPUT_1.ymd;
   im=S_STAB_INPUT_1.im;
   hvd=S_STAB_INPUT_1.hvd;
   lvd=S_STAB_INPUT_1.lvd;
   yvd=S_STAB_INPUT_1.yvd;
   alplov=S_STAB_INPUT_1.alplov;
   clvertmax=S_STAB_INPUT_1.clvertmax;
   qvq=S_STAB_INPUT_1.qvq;
   av=S_STAB_INPUT_1.av;
   hhd=S_STAB_INPUT_1.hhd;
   lhd=S_STAB_INPUT_1.lhd;
   yhd=S_STAB_INPUT_1.yhd;
   alploh=S_STAB_INPUT_1.alploh;
   ih=S_STAB_INPUT_1.ih;
   ah=S_STAB_INPUT_1.ah;
   qhq=S_STAB_INPUT_1.qhq;
   vhv1=S_STAB_INPUT_1.vhv1;
   detafdalpfh=S_STAB_INPUT_1.detafdalpfh;
   htd=S_STAB_INPUT_1.htd;
   ltd=S_STAB_INPUT_1.ltd;
   ytd=S_STAB_INPUT_1.ytd;
   bt=S_STAB_INPUT_1.bt;
   ct=S_STAB_INPUT_1.ct;
   Rt=S_STAB_INPUT_1.Rt;
   at=S_STAB_INPUT_1.at;
   ohmt=S_STAB_INPUT_1.ohmt;
   Ibt=S_STAB_INPUT_1.Ibt;
   delta3=S_STAB_INPUT_1.delta3;
   theta1t=S_STAB_INPUT_1.theta1t;
```

41. uncstructure_stab_input_2.m

```
% Unstructure Consruction for JANRAD98 stability_control_input_2.m

% Written for JANRAD version 6.0 by LT David A. Heathorn

   hwd=S_STAB_INPUT_2.hwd;
   lwd=S_STAB_INPUT_2.lwd;
   ywd=S_STAB_INPUT_2.ywd;
   alplow=S_STAB_INPUT_2.alplow;
   iw=S_STAB_INPUT_2.iw;
   aw=S_STAB_INPUT_2.aw;
   ctw=S_STAB_INPUT_2.ctw;
   crw=S_STAB_INPUT_2.crw;
   vwv1=S_STAB_INPUT_2.vwv1;
   detafdalpfw=S_STAB_INPUT_2.detafdalpfw;
   zcg=S_STAB_INPUT_2.zcg;
   xcg=S_STAB_INPUT_2.xcg;
   ycg=S_STAB_INPUT_2.ycg;
   Ixx=S_STAB_INPUT_2.Ixx;
   Iyy=S_STAB_INPUT_2.Iyy;
   Izz=S_STAB_INPUT_2.Izz;
   Ixz=S_STAB_INPUT_2.Ixz;
   vfv1=S_STAB_INPUT_2.vfv1;
   htnd=S_STAB_INPUT_2.htnd;
   ltnd=S_STAB_INPUT_2.ltnd;
   ytnd=S_STAB_INPUT_2.ytnd;
   dian=S_STAB_INPUT_2.dian;
   swirl=S_STAB_INPUT_2.swirl;
   Ytmaxn=S_STAB_INPUT_2.Ytmaxn;
   lttnd=S_STAB_INPUT_2.lttnd;
   db1mddele=S_STAB_INPUT_2.db1mddele;
   da1mddela=S_STAB_INPUT_2.da1mddela;
   dthetomddelc=S_STAB_INPUT_2.dthetomddelc;
   dthetotddelp=S_STAB_INPUT_2.dthetotddelp;
   sidearm=S_STAB_INPUT_2.sidearm;
   maxr=S_STAB_INPUT_2.maxr;
```

LIST OF REFERENCES

1. Nicholson, R.K. Jr., *Computer code for Interactive Rotorcraft Preliminary Design Using A Harmonic Balance Method for Rotor Trim*, Naval Postgraduate School, thesis for MSAE Degree, Monterey, Ca, 1995.

2. Gerstenberger, W. and Wood, E.R., *Analysis of Helicopter Aeroelastic Characteristics in High-Speed Flight",* AIAA Journal, Volume No. 1, Number 10, Pages 2366-2381, October 1963.

3. Wirth, W.M., *Linear Modeling of Rotorcraft for Stability Analysis and Preliminary Design, Naval Postgraduate School*, Thesis for MSAE Degree, Monterey, CA, 1993.

4. Cuesta J. D., *Modeling Helicopter Blade Dynamics using a Modified Myklestad Prohl Transfer Matrix Method*, Naval Postgraduate School, thesis for MSAE Degree, Monterey, CA, 1994.

5. Hiatt, D. S., *A Study of Helicopter Rotor Dynamics and Modeling Methods*, Naval Postgraduate School, thesis for MSAE Degree, Monterey, CA, 1995.

6. Eccles, D M., A Validation of the Joint Army/Navy rotorcraft Analysis and Design Software by Comparison with H 34 and UUH 60A Flight Test, Naval Postgraduate School, thesis for MSAE Degree, Monterey, CA, 1995.

7. Lapacik, C.F., *Development of Graphical User Interface for Joint Army/Navy Rotorcraft Analysis and Design (JANRAD) Software*, Naval Postgraduate School, Thesis for MSAE Degree, Monterey, CA, 1998.

8. Hucke, W.L., *Performance Enhancements to Joint Army/Navy Rotorcraft Analysis and Design (JANRAD) Software and Graphical User Interface (GUI)*, Naval Postgraduate School, Thesis for MSAE Degree, Monterey, CA, 1998.

9. Prouty, Raymond W., *Helicopter Performance Stability and Control*, Krieger Publishing Company, Malabar Florida, 1995.

10. Amer, Kenneth B. , Gustafson, F.B., "Charts for Estimation of Longitudinal Stability Derivatives for a Helicopter Rotor in Forward Flight", NACA Technical Note 2309, November 25, 1950.

INITIAL DISTRIBUTION LIST

1. Defense Technical Information Center..2
   8725 John J. Kingman Rd., STE 0944
   Ft. Belvoir, Virginia 22060-6218

2. Dudley Knox Library..2
   Naval Postgraduate School
   411 Dyer Rd.
   Monterey, California 93943-5101

3. Professor Gerald Lindsey, Code AA/Li.. 1
   Department of Aeronautics and Astronautics
   Naval Postgraduate School
   Monterey, California 93943-5000

4. Professor E. Roberts Wood, Code AA/Wd..2
   Department of Aeronautics and Astronautics
   Naval Postgraduate School
   Monterey, California 93943-5000

5. LCDR. Robert L. King, USN, Code AA/Ki 1
   Department of Aeronautics and Astronautics
   Naval Postgraduate School
   Monterey, California 93943-5000

6. LT. David A. Heathorn..2
   2416 Steamboat Springs Court
   Chula Vista, California 91955

7. Mr. Raymond Prouty.. 1
   4224 Deerpark Court
   Westlake Village, California 91361

8. Dr. Michael P. Scully 1
   P.O. Box 1120
   Mountain View, California 94042